

# [MS-RDL]:

## Report Definition Language File Format

---

### Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](#) or the [Microsoft Community Promise](#). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **License Programs.** To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](#).
- **Trademarks.** The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](http://www.microsoft.com/trademarks).
- **Fictitious Names.** The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights.** All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

**Support.** For questions and support, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

## Revision Summary

Date	Revision History	Revision Class	Comments
7/13/2009	0.1	Major	Initial availability.
8/7/2009	0.2	Minor	Clarified the meaning of the technical content.
11/6/2009	1.0	Major	Updated and revised the technical content.
3/5/2010	2.0	Major	Updated and revised the technical content.
4/21/2010	2.0.1	Editorial	Changed language and formatting in the technical content.
6/4/2010	2.1	Minor	Clarified the meaning of the technical content.
9/3/2010	2.2	Minor	Clarified the meaning of the technical content.
2/9/2011	3.0	Major	Updated and revised the technical content.
7/7/2011	4.0	Major	Updated and revised the technical content.
11/3/2011	5.0	Major	Updated and revised the technical content.
1/19/2012	5.1	Minor	Clarified the meaning of the technical content.
2/23/2012	5.1	None	No changes to the meaning, language, or formatting of the technical content.
3/27/2012	5.1	None	No changes to the meaning, language, or formatting of the technical content.
5/24/2012	5.1	None	No changes to the meaning, language, or formatting of the technical content.
6/29/2012	5.1	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	6.0	Major	Updated and revised the technical content.
10/8/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.
10/23/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.
3/26/2013	6.1	Minor	Clarified the meaning of the technical content.
6/11/2013	6.2	Minor	Clarified the meaning of the technical content.
8/8/2013	6.2	None	No changes to the meaning, language, or formatting of the technical content.
12/5/2013	6.2	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2014	7.0	Major	Updated and revised the technical content.
5/20/2014	8.0	Major	Updated and revised the technical content.
5/10/2016	9.0	Major	Significantly changed the technical content.
8/16/2017	10.0	Major	Significantly changed the technical content.



<b>Date</b>	<b>Revision History</b>	<b>Revision Class</b>	<b>Comments</b>
10/16/2019	11.0	Major	Significantly changed the technical content.
12/18/2019	11.1	Minor	Clarified the meaning of the technical content.
6/22/2020	11.2	Minor	Clarified the meaning of the technical content.
7/30/2020	11.2	None	No changes to the meaning, language, or formatting of the technical content.
11/1/2022	12.0	Major	Significantly changed the technical content.
11/30/2022	13.0	Major	Significantly changed the technical content.
3/8/2023	14.0	Major	Significantly changed the technical content.

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>55</b>
1.1	Glossary .....	55
1.2	References .....	61
1.2.1	Normative References .....	61
1.2.2	Informative References .....	62
1.3	Overview .....	62
1.3.1	Report Definition Language File Content .....	62
1.3.2	Report Rendering.....	63
1.3.3	Document Structure.....	63
1.3.3.1	XML Namespace .....	63
1.3.4	Report Definition Overview Diagrams .....	64
1.4	Relationship to Protocols and Other Structures .....	70
1.5	Applicability Statement .....	71
1.6	Versioning and Localization .....	71
1.7	Vendor-Extensible Fields .....	71
<b>2</b>	<b>Structures .....</b>	<b>72</b>
2.1	Introduction .....	72
2.2	Common RDL Types .....	73
2.2.1	String.....	73
2.2.2	Integer.....	73
2.2.3	Boolean.....	73
2.2.4	Float .....	74
2.2.5	DateTime.....	74
2.2.6	NormalizedString Elements and Attributes .....	74
2.2.7	RdISize.....	76
2.2.8	RdIColor .....	78
2.2.9	RdIURL .....	84
2.2.10	ReportMIMEType .....	84
2.2.11	ReportPath .....	85
2.2.12	ReportLanguage .....	86
2.2.13	Complex Types.....	86
2.2.14	StringWithDataModelAttribute.....	86
2.2.15	LocIDStringWithDataModelAttribute.....	87
2.2.16	StringWithValueModelAttribute .....	88
2.3	Report .....	89
2.3.1	Report.MustUnderstand .....	94
2.3.2	Report.AuthoringMetadata .....	94
2.3.3	Report.Author .....	95
2.3.4	Report.AutoRefresh.....	95
2.3.5	Report.Body .....	96
2.3.6	Report.Classes .....	96
2.3.7	Report.Code.....	96
2.3.8	Report.CodeModules .....	96
2.3.9	Report.ConsumeContainerWhitespace .....	97
2.3.10	Report.CustomProperties .....	97
2.3.11	Report.DataElementName.....	98
2.3.12	Report.DataElementStyle.....	98
2.3.13	Report.DataSchema .....	99
2.3.14	Report.DataSets .....	99
2.3.15	Report.DataSources .....	99
2.3.16	Report.DataTransform .....	99
2.3.17	Report.DefaultFontFamily .....	100
2.3.18	Report.DeferVariableEvaluation .....	100
2.3.19	Report.Description .....	101

2.3.20	Report.EmbeddedImages.....	101
2.3.21	Report.InitialPageName .....	101
2.3.22	Report.InteractiveHeight .....	101
2.3.23	Report.InteractiveWidth .....	102
2.3.24	Report.Language .....	102
2.3.25	Report.BottomMargin .....	102
2.3.26	Report.LeftMargin .....	103
2.3.27	Report.TopMargin .....	103
2.3.28	Report.RightMargin .....	103
2.3.29	Report.PageFooter .....	104
2.3.30	Report.PageHeader .....	104
2.3.31	Report.PageHeight .....	104
2.3.32	Report.PageWidth .....	105
2.3.33	Report.Page .....	105
2.3.34	Report.ReportParameters .....	105
2.3.35	Report.ReportParametersLayout .....	106
2.3.36	Report.Variables .....	106
2.3.37	Report.Width.....	106
2.3.38	Report.ReportSections.....	107
2.4	AuthoringMetadata .....	107
2.4.1	AuthoringMetadata.CreatedBy .....	108
2.4.2	AuthoringMetadata.UpdatedBy .....	108
2.4.3	AuthoringMetadata.Name .....	108
2.4.4	AuthoringMetadata.Version .....	109
2.5	ReportSections .....	109
2.5.1	ReportSections.ReportSection .....	110
2.6	ReportSection .....	110
2.6.1	ReportSection.Body .....	111
2.6.2	ReportSection.Page.....	111
2.6.3	ReportSection.Width .....	112
2.6.4	ReportSection.DataElementName .....	112
2.6.5	ReportSection.DataElementOutput.....	112
2.6.6	ReportSection.Name .....	113
2.6.7	ReportSection.LayoutDirection .....	113
2.7	Body .....	114
2.7.1	Body.Columns .....	115
2.7.2	Body.ColumnSpacing .....	115
2.7.3	Body.Style .....	116
2.7.4	Body.Height .....	116
2.7.5	Body.ReportItems.....	116
2.8	Page.....	117
2.8.1	Page.Columns .....	118
2.8.2	Page.ColumnSpacing.....	118
2.8.3	Page.BottomMargin.....	119
2.8.4	Page.InteractiveHeight .....	119
2.8.5	Page.InteractiveWidth .....	119
2.8.6	Page.LeftMargin.....	120
2.8.7	Page.PageFooter.....	120
2.8.8	Page.PageHeader.....	120
2.8.9	Page.PageHeight .....	121
2.8.10	Page.PageWidth .....	121
2.8.11	Page.RightMargin .....	121
2.8.12	Page.Style .....	122
2.8.13	Page.TopMargin.....	122
2.9	PageSection .....	122
2.9.1	PageSection.Style .....	124
2.9.2	PageSection.Height.....	124
2.9.3	PageSection.PrintOnFirstPage.....	124

2.9.4	PageSection.PrintOnLastPage	125
2.9.5	PageSection.PrintBetweenSections	125
2.9.6	PageSection.ReportItems	125
2.10	PageHeaderFooter	126
2.10.1	PageHeaderFooter.Height	127
2.10.2	PageHeaderFooter.PrintOnFirstPage	127
2.10.3	PageHeaderFooter.PrintOnLastPage	127
2.10.4	PageHeaderFooter.ReportItems	128
2.10.5	PageHeaderFooter.Style	128
2.11	ReportItems	128
2.11.1	ReportItems.Chart	130
2.11.2	ReportItems.CustomReportItem	131
2.11.3	ReportItems.GaugePanel	131
2.11.4	ReportItems.Image	131
2.11.5	ReportItems.Line	132
2.11.6	ReportItems.Map	132
2.11.7	ReportItems.Rectangle	132
2.11.8	ReportItems.Subreport	133
2.11.9	ReportItems.Tablix	133
2.11.10	ReportItems.Textbox	133
2.11.11	ReportItems.List	133
2.11.12	ReportItems.Matrix	134
2.11.13	ReportItems.Table	134
2.12	Image	134
2.12.1	Image.Name	137
2.12.2	Image.Style	137
2.12.3	Image.ActionInfo	137
2.12.4	Image.Bookmark	138
2.12.5	Image.CustomProperties	138
2.12.6	Image.DataElementName	138
2.12.7	Image.DataElementOutput	139
2.12.8	Image.DocumentMapLabel	139
2.12.9	Image.Height	139
2.12.10	Image.Left	140
2.12.11	Image.RepeatWith	140
2.12.12	Image.ToolTip	140
2.12.13	Image.Top	141
2.12.14	Image.Visibility	141
2.12.15	Image.Width	141
2.12.16	Image.ZIndex	142
2.12.17	Image.MIMEType	142
2.12.18	Image.Sizing	142
2.12.19	Image.Source	143
2.12.20	Image.Value	144
2.12.21	Image.Tag	144
2.12.22	Image.Tags	144
2.12.23	Image.EmbeddingMode	145
2.13	Line	146
2.13.1	Line.Name	147
2.13.2	Line.Style	147
2.13.3	Line.ActionInfo	148
2.13.4	Line.Bookmark	148
2.13.5	Line.CustomProperties	148
2.13.6	Line.DataElementName	149
2.13.7	Line.DataElementOutput	149
2.13.8	Line.DocumentMapLabel	149
2.13.9	Line.Height	150
2.13.10	Line.Left	150

2.13.11	Line.RepeatWith .....	150
2.13.12	Line.ToolTip .....	151
2.13.13	Line.Top .....	151
2.13.14	Line.Visibility.....	151
2.13.15	Line.Width .....	152
2.13.16	Line.ZIndex .....	152
2.14	Rectangle.....	152
2.14.1	Rectangle.Name .....	155
2.14.2	Rectangle.Style .....	156
2.14.3	Rectangle.ActionInfo .....	156
2.14.4	Rectangle.Bookmark .....	156
2.14.5	Rectangle.CustomProperties.....	157
2.14.6	Rectangle.DataElementName .....	157
2.14.7	Rectangle.DataElementOutput.....	157
2.14.8	Rectangle.DocumentMapLabel .....	158
2.14.9	Rectangle.Height .....	158
2.14.10	Rectangle.Left .....	159
2.14.11	Rectangle.RepeatWith .....	159
2.14.12	Rectangle.ToolTip .....	159
2.14.13	Rectangle.Top .....	160
2.14.14	Rectangle.Visibility.....	160
2.14.15	Rectangle.Width .....	160
2.14.16	Rectangle.ZIndex .....	161
2.14.17	Rectangle.KeepTogether .....	161
2.14.18	Rectangle.LinkToChild .....	161
2.14.19	Rectangle.OmitBorderOnPageBreak .....	162
2.14.20	Rectangle.PageBreak .....	162
2.14.21	Rectangle.ReportItems .....	162
2.14.22	Rectangle.PageBreakAtEnd .....	163
2.14.23	Rectangle.PageBreakAtStart.....	163
2.14.24	Rectangle.PageName .....	163
2.15	Subreport .....	164
2.15.1	Subreport.Name .....	167
2.15.2	Subreport.Style .....	167
2.15.3	Subreport.ActionInfo .....	168
2.15.4	Subreport.Bookmark .....	168
2.15.5	Subreport.CustomProperties .....	168
2.15.6	Subreport.DataElementName .....	168
2.15.7	Subreport.DataElementOutput .....	169
2.15.8	Subreport.DocumentMapLabel.....	169
2.15.9	Subreport.Height .....	170
2.15.10	Subreport.Left.....	170
2.15.11	Subreport.LinkToChild .....	170
2.15.12	Subreport.RepeatWith .....	171
2.15.13	Subreport.ToolTip .....	171
2.15.14	Subreport.Top .....	171
2.15.15	Subreport.Visibility .....	172
2.15.16	Subreport.Width .....	172
2.15.17	Subreport.ZIndex .....	172
2.15.18	Subreport.KeepTogether.....	173
2.15.19	Subreport.MergeTransactions .....	173
2.15.20	Subreport.NoRowsMessage .....	173
2.15.21	Subreport.OmitBorderOnPageBreak .....	174
2.15.22	Subreport.Parameters .....	174
2.15.23	Subreport.ReportName.....	174
2.15.24	Subreport.NoRows .....	175
2.16	Textbox .....	175
2.16.1	Textbox.Name.....	179

2.16.2	Textbox.Style	179
2.16.3	Textbox.ActionInfo	179
2.16.4	Textbox.Bookmark	180
2.16.5	Textbox.CanScrollVertically	180
2.16.6	Textbox.CustomProperties	180
2.16.7	Textbox.DataElementName	181
2.16.8	Textbox.DataElementOutput	181
2.16.9	Textbox.DocumentMapLabel	182
2.16.10	Textbox.Height	182
2.16.11	Textbox.Left	182
2.16.12	Textbox.RepeatWith	183
2.16.13	Textbox.ToolTip	183
2.16.14	Textbox.Top	183
2.16.15	Textbox.Visibility	184
2.16.16	Textbox.Width	184
2.16.17	Textbox.ZIndex	184
2.16.18	Textbox.CanGrow	185
2.16.19	Textbox.CanShrink	185
2.16.20	Textbox.DataElementStyle	185
2.16.21	Textbox.HideDuplicates	186
2.16.22	Textbox.KeepTogether	187
2.16.23	Textbox.Paragraphs	187
2.16.24	Textbox.ToggleImage	187
2.16.25	Textbox.UserSort	188
2.16.26	Textbox.Value	188
2.17	Paragraphs	188
2.17.1	Paragraphs.Paragraph	189
2.18	Paragraph	189
2.18.1	Paragraph.HangingIndent	191
2.18.2	Paragraph.LeftIndent	191
2.18.3	Paragraph.ListLevel	191
2.18.4	Paragraph.ListStyle	192
2.18.5	Paragraph.RightIndent	192
2.18.6	Paragraph.SpaceAfter	193
2.18.7	Paragraph.SpaceBefore	193
2.18.8	Paragraph.Style	193
2.18.9	Paragraph.TextRuns	194
2.19	TextRuns	194
2.19.1	TextRuns.TextRun	195
2.20	TextRun	195
2.20.1	TextRun.ActionInfo	196
2.20.2	TextRun.Label	196
2.20.3	TextRun.MarkupType	197
2.20.4	TextRun.Style	197
2.20.5	TextRun.ToolTip	197
2.20.6	TextRun.Value	198
2.20.7	TextRun.Value.DataType	198
2.20.8	TextRun.Value.EvaluationMode	199
2.21	ToggleImage	199
2.21.1	ToggleImage.InitialState	200
2.22	UserSort	200
2.22.1	UserSort.SortExpression	201
2.22.2	UserSort.SortExpressionScope	201
2.22.3	UserSort.SortTarget	202
2.23	Tablix	202
2.23.1	Tablix.Name	207
2.23.2	Tablix.ActionInfo	207
2.23.3	Tablix.BandLayoutOptions	208

2.23.4	Tablix.BottomMargin	208
2.23.5	Tablix.Bookmark	208
2.23.6	Tablix.CanScroll	209
2.23.7	Tablix.CustomProperties	209
2.23.8	Tablix.DataElementName	209
2.23.9	Tablix.DataElementOutput	210
2.23.10	Tablix.DataSetName	210
2.23.11	Tablix.DocumentMapLabel	211
2.23.12	Tablix.Filters	211
2.23.13	Tablix.FixedColumnHeaders	212
2.23.14	Tablix.FixedRowHeaders	212
2.23.15	Tablix.GroupsBeforeRowHeaders	212
2.23.16	Tablix.Height	213
2.23.17	Tablix.KeepTogether	214
2.23.18	Tablix.LayoutDirection	214
2.23.19	Tablix.Left	215
2.23.20	Tablix.LeftMargin	215
2.23.21	Tablix.NoRowsMessage	215
2.23.22	Tablix.OmitBorderOnPageBreak	216
2.23.23	Tablix.PageBreak	216
2.23.24	Tablix.PageName	216
2.23.25	Tablix.Relationship	217
2.23.26	Tablix.RepeatColumnHeaders	217
2.23.27	Tablix.RepeatRowHeaders	218
2.23.28	Tablix.RepeatWith	218
2.23.29	Tablix.ReportSlicerState	218
2.23.30	Tablix.RightMargin	218
2.23.31	Tablix.SortExpressions	219
2.23.32	Tablix.Style	219
2.23.33	Tablix.TablixBody	219
2.23.34	Tablix.TablixColumnHierarchy	220
2.23.35	Tablix.TablixCorner	220
2.23.36	Tablix.TablixRowHierarchy	221
2.23.37	Tablix.ToolTip	221
2.23.38	Tablix.Top	221
2.23.39	Tablix.TopMargin	222
2.23.40	Tablix.Visibility	222
2.23.41	Tablix.Width	222
2.23.42	Tablix.ZIndex	223
2.24	TablixBody	223
2.24.1	TablixBody.TablixColumns	224
2.24.2	TablixBody.TablixRows	224
2.25	TablixColumns	224
2.25.1	TablixColumns.TablixColumn	225
2.26	TablixColumn	225
2.26.1	TablixColumn.Width	226
2.27	TablixRows	226
2.27.1	TablixRows.TablixRow	227
2.28	TablixRow	228
2.28.1	TablixRow.Height	228
2.28.2	TablixRow.TablixCells	229
2.29	TablixCells	229
2.29.1	TablixCells.TablixCell	230
2.30	TablixCell	230
2.30.1	TablixCell.CellContents	231
2.30.2	TablixCell.DataElementName	232
2.30.3	TablixCell.DataElementOutput	232
2.30.4	TablixCell.DataSetName	233

2.30.5	TablixCell.Relationships .....	233
2.31	CellContents .....	234
2.31.1	CellContents.Chart .....	235
2.31.2	CellContents.ColSpan .....	236
2.31.3	CellContents.CustomReportItem .....	237
2.31.4	CellContents.GaugePanel .....	237
2.31.5	CellContents.Image .....	238
2.31.6	CellContents.Line .....	238
2.31.7	CellContents.Map .....	239
2.31.8	CellContents.Rectangle .....	239
2.31.9	CellContents.RowSpan .....	240
2.31.10	CellContents.Subreport .....	240
2.31.11	CellContents.Tablix .....	241
2.31.12	CellContents.Textbox .....	241
2.32	TablixColumnHierarchy .....	242
2.32.1	TablixColumnHierarchy.TablixMembers .....	243
2.32.2	TablixColumnHierarchy.EnableDrilldown .....	243
2.33	TablixMembers .....	243
2.33.1	TablixMembers.TablixMember .....	244
2.34	TablixMember .....	244
2.34.1	TablixMember.CustomProperties .....	246
2.34.2	TablixMember.DataElementName .....	247
2.34.3	TablixMember.DataElementOutput .....	247
2.34.4	TablixMember.FixedData .....	248
2.34.5	TablixMember.Group .....	249
2.34.6	TablixMember.HideIfNoRows .....	249
2.34.7	TablixMember.KeepTogether .....	250
2.34.8	TablixMember.KeepWithGroup .....	250
2.34.9	TablixMember.RepeatOnNewPage .....	251
2.34.10	TablixMember.SortExpressions .....	252
2.34.11	TablixMember.TablixHeader .....	252
2.34.12	TablixMember.TablixMembers .....	253
2.34.13	TablixMember.Visibility .....	253
2.35	TablixHeader .....	253
2.35.1	TablixHeader.CellContents .....	254
2.35.2	TablixHeader.Size .....	255
2.36	TablixCorner .....	255
2.36.1	TablixCorner.TablixCornerRows .....	256
2.37	TablixCornerRows .....	256
2.37.1	TablixCornerRows.TablixCornerRow .....	257
2.38	TablixCornerRow .....	257
2.38.1	TablixCornerRow.TablixCornerCell .....	258
2.39	TablixCornerCell .....	259
2.39.1	TablixCornerCell.CellContents .....	259
2.40	TablixHierarchy .....	260
2.40.1	TablixHierarchy.TablixMembers .....	261
2.41	TablixRowHierarchy .....	261
2.41.1	TablixRowHierarchy.TablixMembers .....	262
2.41.2	TablixRowHierarchy.EnableDrilldown .....	262
2.42	BandLayoutOptions .....	263
2.42.1	BandLayoutOptions.RowCount .....	264
2.42.2	BandLayoutOptions.ColumnCount .....	264
2.42.3	BandLayoutOptions.Coverflow .....	265
2.42.4	BandLayoutOptions.PlayAxis .....	265
2.42.5	BandLayoutOptions.Tabstrip .....	265
2.43	Coverflow .....	266
2.43.1	Coverflow.NavigationItem .....	266
2.43.2	Coverflow.Slider .....	267



2.44	Tabstrip .....	267
2.44.1	Tabstrip.NavigationItem .....	268
2.44.2	Tabstrip.Slider.....	268
2.45	PlayAxis .....	268
2.45.1	PlayAxis.Slider .....	269
2.45.2	PlayAxis.DockingOption .....	269
2.46	NavigationItem .....	270
2.46.1	NavigationItem.ReportItemReference.....	270
2.46.2	NavigationItem.ReportItem.....	271
2.47	Slider .....	271
2.47.1	Slider.Hidden .....	272
2.47.2	Slider.LabelData .....	272
2.48	LabelData .....	272
2.48.1	LabelData.DataSetName.....	273
2.48.2	LabelData.Key.....	274
2.48.3	LabelData.KeyFields.....	274
2.48.4	LabelData.Label.....	275
2.49	List.....	275
2.49.1	List.Name .....	277
2.49.2	List.Style .....	278
2.49.3	List.Action .....	278
2.49.4	List.LinkToChild .....	278
2.49.5	List.Bookmark .....	278
2.49.6	List.CustomProperties .....	279
2.49.7	List.DataElementName .....	279
2.49.8	List.DataElementOutput.....	280
2.49.9	List.Label.....	281
2.49.10	List.Height .....	281
2.49.11	List.Left.....	281
2.49.12	List.RepeatWith .....	282
2.49.13	List.ToolTip .....	282
2.49.14	List.Top .....	282
2.49.15	List.Visibility .....	283
2.49.16	List.Width .....	283
2.49.17	List.ZIndex .....	283
2.49.18	List.KeepTogether.....	284
2.49.19	List.NoRows .....	284
2.49.20	List.PageBreakAtEnd .....	284
2.49.21	List.PageBreakAtStart .....	285
2.49.22	List.DataSetName.....	285
2.49.23	List.Filters.....	286
2.49.24	List.DataInstanceElementOutput.....	286
2.49.25	List.DataInstanceName .....	287
2.49.26	List.FillPage .....	287
2.49.27	List.Grouping .....	287
2.49.28	List.ReportItems.....	288
2.49.29	List.Sorting .....	288
2.50	Matrix.....	288
2.50.1	Matrix.Name .....	291
2.50.2	Matrix.Style .....	291
2.50.3	Matrix.Action.....	292
2.50.4	Matrix.LinkToChild .....	292
2.50.5	Matrix.Bookmark .....	292
2.50.6	Matrix.CustomProperties.....	293
2.50.7	Matrix.DataElementName .....	293
2.50.8	Matrix.DataElementOutput.....	293
2.50.9	Matrix.Label .....	294
2.50.10	Matrix.Height .....	295

2.50.11	Matrix.Left	295
2.50.12	Matrix.RepeatWith	296
2.50.13	Matrix.ToolTip	296
2.50.14	Matrix.Top	296
2.50.15	Matrix.Visibility	297
2.50.16	Matrix.Width	297
2.50.17	Matrix.ZIndex	297
2.50.18	Matrix.KeepTogether	298
2.50.19	Matrix.NoRows	298
2.50.20	Matrix.PageBreakAtEnd	298
2.50.21	Matrix.PageBreakAtStart	299
2.50.22	Matrix.DataSetName	299
2.50.23	Matrix.Filters	300
2.50.24	Matrix.CellDataElementName	300
2.50.25	Matrix.CellDataElementOutput	300
2.50.26	Matrix.ColumnGroupings	301
2.50.27	Matrix.Corner	301
2.50.28	Matrix.GroupsBeforeRowHeaders	302
2.50.29	Matrix.LayoutDirection	302
2.50.30	Matrix.MatrixColumns	303
2.50.31	Matrix.MatrixRows	303
2.50.32	Matrix.RowGroupings	303
2.51	ColumnGroupings	304
2.51.1	ColumnGroupings.ColumnGrouping	304
2.52	ColumnGrouping	304
2.52.1	ColumnGrouping.DynamicColumns	305
2.52.2	ColumnGrouping.FixedHeader	305
2.52.3	ColumnGrouping.Height	306
2.52.4	ColumnGrouping.StaticColumns	306
2.53	DynamicColumns	307
2.53.1	DynamicColumns.Grouping	307
2.53.2	DynamicColumns.ReportItems	308
2.53.3	DynamicColumns.Sorting	308
2.53.4	DynamicColumns.Subtotal	308
2.53.5	DynamicColumns.Visibility	309
2.54	Subtotal	309
2.54.1	Subtotal.DataElementName	310
2.54.2	Subtotal.DataElementOutput	310
2.54.3	Subtotal.Position	311
2.54.4	Subtotal.ReportItems	311
2.54.5	Subtotal.Style	312
2.55	StaticColumns	312
2.55.1	StaticColumns.StaticColumn	313
2.56	StaticColumn	313
2.56.1	StaticColumn.ReportItems	314
2.57	Corner	314
2.57.1	Corner.ReportItems	315
2.58	MatrixColumns	315
2.58.1	MatrixColumns.MatrixColumn	316
2.59	MatrixColumn	316
2.59.1	MatrixColumn.Width	316
2.60	MatrixRows	317
2.60.1	MatrixRows.MatrixRow	317
2.61	MatrixRow	318
2.61.1	MatrixRow.Height	318
2.61.2	MatrixRow.MatrixCells	318
2.62	MatrixCells	319
2.62.1	MatrixCells.MatrixCell	319

2.63	MatrixCell.....	320
2.63.1	MatrixCell.ReportItems.....	320
2.64	RowGroupings .....	321
2.64.1	RowGroupings.RowGrouping .....	321
2.65	RowGrouping.....	322
2.65.1	RowGrouping.DynamicRows.....	322
2.65.2	RowGrouping.FixedHeader .....	323
2.65.3	RowGrouping.StaticRows .....	323
2.65.4	RowGrouping.Width .....	323
2.66	DynamicRows .....	324
2.66.1	DynamicRows.Grouping.....	324
2.66.2	DynamicRows.ReportItems .....	325
2.66.3	DynamicRows.Sorting .....	325
2.66.4	DynamicRows.Subtotal.....	325
2.66.5	DynamicRows.Visibility .....	326
2.67	StaticRows .....	326
2.67.1	StaticRows.StaticRow.....	327
2.68	StaticRow.....	327
2.68.1	StaticRow.ReportItems.....	327
2.69	Table.....	328
2.69.1	Table.Name .....	330
2.69.2	Table.Style .....	330
2.69.3	Table.Action.....	331
2.69.4	Table.LinkToChild .....	331
2.69.5	Table.Bookmark .....	331
2.69.6	Table.CustomProperties.....	332
2.69.7	Table.DataElementName .....	332
2.69.8	Table.DataElementOutput.....	332
2.69.9	Table.Label .....	333
2.69.10	Table.Height .....	334
2.69.11	Table.Left .....	334
2.69.12	Table.RepeatWith .....	335
2.69.13	Table.ToolTip .....	335
2.69.14	Table.Top .....	335
2.69.15	Table.Visibility.....	336
2.69.16	Table.Width .....	336
2.69.17	Table.ZIndex.....	336
2.69.18	Table.KeepTogether .....	337
2.69.19	Table.NoRows .....	337
2.69.20	Table.PageBreakAtEnd .....	337
2.69.21	Table.PageBreakAtStart.....	338
2.69.22	Table.DataSetName .....	338
2.69.23	Table.Filters .....	338
2.69.24	Table.DetailDataCollectionName .....	339
2.69.25	Table.DetailDataElementName .....	339
2.69.26	Table.DetailDataElementOutput.....	340
2.69.27	Table.Details.....	340
2.69.28	Table.FillPage.....	341
2.69.29	Table.Footer.....	341
2.69.30	Table.Header .....	341
2.69.31	Table.TableColumns .....	342
2.69.32	Table.TableGroups .....	342
2.70	Details.....	342
2.70.1	Details.Grouping.....	343
2.70.2	Details.Sorting .....	343
2.70.3	Details.TableRows.....	344
2.70.4	Details.Visibility .....	344
2.71	TableRows.....	344

2.71.1	TableRows.TableRow	345
2.72	TableRow	345
2.72.1	TableRow.Height	346
2.72.2	TableRow.TableCells	346
2.72.3	TableRow.Visibility	346
2.73	TableCells	347
2.73.1	TableCells.TableCell	347
2.74	TableCell	348
2.74.1	TableCell.ColSpan	348
2.74.2	TableCell.ReportItems	349
2.75	Footer	349
2.75.1	Footer.RepeatOnNewPage	350
2.75.2	Footer.TableRows	350
2.76	Header	350
2.76.1	Header.FixedHeader	351
2.76.2	Header.RepeatOnNewPage	351
2.76.3	Header.TableRows	352
2.77	TableColumns	352
2.77.1	TableColumns.TableColumn	353
2.78	TableColumn	353
2.78.1	TableColumn.FixedHeader	353
2.78.2	TableColumn.Visibility	354
2.78.3	TableColumn.Width	354
2.79	TableGroups	354
2.79.1	TableGroups.TableGroup	355
2.80	TableGroup	355
2.80.1	TableGroup.Footer	356
2.80.2	TableGroup.Grouping	356
2.80.3	TableGroup.Header	357
2.80.4	TableGroup.Sorting	357
2.80.5	TableGroup.Visibility	357
2.81	Group	358
2.81.1	Group.Name	360
2.81.2	Group.DataElementName	360
2.81.3	Group.DataElementOutput	360
2.81.4	Group.DataSetName	361
2.81.5	Group.DocumentMapLabel	361
2.81.6	Group.DomainScope	362
2.81.7	Group.Filters	362
2.81.8	Group.GroupExpressions	362
2.81.9	Group.NaturalGroup	363
2.81.10	Group.PageBreak	363
2.81.11	Group.PageName	363
2.81.12	Group.Parent	364
2.81.13	Group.ReGroupExpressions	364
2.81.14	Group.Relationship	364
2.81.15	Group.Variables	365
2.82	GroupExpressions	365
2.82.1	GroupExpressions.GroupExpression	366
2.83	SortExpressions	366
2.83.1	SortExpressions.SortExpression	367
2.84	SortExpression	367
2.84.1	SortExpression.Direction	368
2.84.2	SortExpression.NaturalSort	369
2.84.3	SortExpression.Value	369
2.84.4	SortExpression.DeferredSort	370
2.85	Grouping	370
2.85.1	Grouping.Name	371

2.85.2	Grouping.CustomProperties .....	372
2.85.3	Grouping.DataCollectionName .....	372
2.85.4	Grouping.DataElementName .....	373
2.85.5	Grouping.DataElementOutput.....	373
2.85.6	Grouping.Filters.....	374
2.85.7	Grouping.GroupExpressions .....	374
2.85.8	Grouping.Label.....	374
2.85.9	Grouping.PageBreakAtEnd .....	375
2.85.10	Grouping.PageBreakAtStart .....	375
2.85.11	Grouping.Parent .....	375
2.86	Sorting .....	376
2.86.1	Sorting.SortBy .....	377
2.87	SortBy .....	377
2.87.1	SortBy.Direction .....	378
2.87.2	SortBy.SortExpression.....	378
2.88	Chart.....	378
2.88.1	Chart.Name .....	385
2.88.2	Chart.Action.....	385
2.88.3	Chart.ActionInfo .....	385
2.88.4	Chart.Bookmark .....	386
2.88.5	Chart.CategoryAxis .....	386
2.88.6	Chart.CategoryGroupings .....	386
2.88.7	Chart.ChartAnnotations .....	387
2.88.8	Chart.ChartAreas.....	387
2.88.9	Chart.ChartBorderSkin .....	387
2.88.10	Chart.ChartCodeParameters.....	387
2.88.11	Chart.ChartCategoryHierarchy.....	388
2.88.12	Chart.ChartCustomPaletteColors.....	388
2.88.13	Chart.ChartData .....	389
2.88.14	Chart.ChartElementOutput.....	389
2.88.15	Chart.ChartLegends .....	389
2.88.16	Chart.ChartNoDataMessage .....	390
2.88.17	Chart.ChartSeriesHierarchy.....	390
2.88.18	Chart.ChartTitles .....	390
2.88.19	Chart.Code .....	391
2.88.20	Chart.CodeLanguage.....	391
2.88.21	Chart.CustomProperties.....	391
2.88.22	Chart.DataElementName .....	392
2.88.23	Chart.DataElementOutput.....	392
2.88.24	Chart.DataSetName .....	393
2.88.25	Chart.DocumentMapLabel .....	393
2.88.26	Chart.DynamicHeight .....	393
2.88.27	Chart.DynamicWidth .....	394
2.88.28	Chart.Filters.....	394
2.88.29	Chart.Height .....	394
2.88.30	Chart.KeepTogether.....	395
2.88.31	Chart.Label .....	395
2.88.32	Chart.Left .....	395
2.88.33	Chart.Legend .....	396
2.88.34	Chart.LinkToChild .....	396
2.88.35	Chart.NoRows .....	396
2.88.36	Chart.NoRowsMessage .....	397
2.88.37	Chart.PageBreak.....	397
2.88.38	Chart.PageBreakAtEnd .....	397
2.88.39	Chart.PageBreakAtStart.....	398
2.88.40	Chart.PageName .....	398
2.88.41	Chart.Palette.....	398
2.88.42	Chart.PaletteHatchBehavior .....	400

2.88.43	Chart.PlotArea	400
2.88.44	Chart.PointWidth	400
2.88.45	Chart.Relationship	401
2.88.46	Chart.RepeatWith	401
2.88.47	Chart.SeriesGroupings	402
2.88.48	Chart.SortExpressions	402
2.88.49	Chart.Style	402
2.88.50	Chart.Subtype	402
2.88.51	Chart.ThreeDProperties	403
2.88.52	Chart.Title	404
2.88.53	Chart.ToolTip	404
2.88.54	Chart.Top	404
2.88.55	Chart.Type	405
2.88.56	Chart.ValueAxis	406
2.88.57	Chart.Visibility	406
2.88.58	Chart.Width	406
2.88.59	Chart.ZIndex	406
2.89	CategoryAxis	407
2.89.1	CategoryAxis.Axis	407
2.90	Axis	408
2.90.1	Axis.CrossAt	409
2.90.2	Axis.Interlaced	409
2.90.3	Axis.LogScale	410
2.90.4	Axis.MajorGridLines	410
2.90.5	Axis.MajorInterval	411
2.90.6	Axis.MajorTickMarks	411
2.90.7	Axis.Margin	412
2.90.8	Axis.Max	412
2.90.9	Axis.Min	412
2.90.10	Axis.MinorGridLines	413
2.90.11	Axis.MinorInterval	413
2.90.12	Axis.MinorTickMarks	413
2.90.13	Axis.Reverse	414
2.90.14	Axis.Scalar	414
2.90.15	Axis.Style	415
2.90.16	Axis.Title	415
2.90.17	Axis.Visible	415
2.91	MajorGridLines	416
2.91.1	MajorGridLines.ShowGridLines	416
2.91.2	MajorGridLines.Style	417
2.92	MinorGridLines	417
2.92.1	MinorGridLines.ShowGridLines	417
2.92.2	MinorGridLines.Style	418
2.93	CategoryGroupings	418
2.93.1	CategoryGroupings.CategoryGrouping	419
2.94	CategoryGrouping	419
2.94.1	CategoryGrouping.DynamicCategories	420
2.94.2	CategoryGrouping.StaticCategories	420
2.95	DynamicCategories	420
2.95.1	DynamicCategories.Grouping	421
2.95.2	DynamicCategories.Label	421
2.95.3	DynamicCategories.Sorting	422
2.96	StaticCategories	422
2.96.1	StaticCategories.StaticMember	422
2.97	StaticMember	423
2.97.1	StaticMember.Label	423
2.98	ChartData	424
2.98.1	ChartData.ChartSeries	425

2.98.2	ChartData.ChartSeriesCollection .....	425
2.98.3	ChartData.ChartDerivedSeriesCollection .....	425
2.99	ChartSeries .....	426
2.99.1	ChartSeries.Name .....	428
2.99.2	ChartSeries.CategoryAxisName .....	429
2.99.3	ChartSeries.ChartAreaName .....	429
2.99.4	ChartSeries.ChartDataLabel .....	429
2.99.5	ChartSeries.ChartDataPoints .....	430
2.99.6	ChartSeries.ChartEmptyPoints .....	430
2.99.7	ChartSeries.ChartItemInLegend .....	430
2.99.8	ChartSeries.ChartMarker .....	431
2.99.9	ChartSeries.ChartSmartLabel .....	431
2.99.10	ChartSeries.CustomProperties .....	431
2.99.11	ChartSeries.DataPoints .....	432
2.99.12	ChartSeries.Hidden .....	432
2.99.13	ChartSeries.LegendName .....	432
2.99.14	ChartSeries.PlotType .....	433
2.99.15	ChartSeries.Style .....	433
2.99.16	ChartSeries.Subtype .....	434
2.99.17	ChartSeries.Type .....	435
2.99.18	ChartSeries.ValueAxisName .....	436
2.100	DataPoints .....	436
2.100.1	DataPoints.DataPoint .....	437
2.101	DataPoint .....	437
2.101.1	DataPoint.Action .....	438
2.101.2	DataPoint.DataElementName .....	438
2.101.3	DataPoint.DataElementOutput .....	439
2.101.4	DataPoint.DataLabel .....	439
2.101.5	DataPoint.DataValues .....	440
2.101.6	DataPoint.Marker .....	440
2.101.7	DataPoint.Style .....	440
2.102	DataLabel .....	441
2.102.1	DataLabel.Position .....	442
2.102.2	DataLabel.Rotation .....	443
2.102.3	DataLabel.Style .....	443
2.102.4	DataLabel.Value .....	443
2.102.5	DataLabel.Visible .....	444
2.103	DataValues .....	444
2.103.1	DataValues.DataValue .....	444
2.104	Marker .....	445
2.104.1	Marker.Size .....	445
2.104.2	Marker.Style .....	446
2.104.3	Marker.Type .....	446
2.105	Legend .....	447
2.105.1	Legend.InsidePlotArea .....	448
2.105.2	Legend.Layout .....	448
2.105.3	Legend.Position .....	449
2.105.4	Legend.Visible .....	450
2.106	PlotArea .....	450
2.106.1	PlotArea.Style .....	451
2.107	SeriesGroupings .....	451
2.107.1	SeriesGroupings.SeriesGrouping .....	451
2.108	SeriesGrouping .....	452
2.108.1	SeriesGrouping.DynamicSeries .....	452
2.108.2	SeriesGrouping.StaticSeries .....	453
2.109	DynamicSeries .....	453
2.109.1	DynamicSeries.Grouping .....	454
2.109.2	DynamicSeries.Label .....	454

2.109.3	DynamicSeries.Sorting	454
2.110	StaticSeries	455
2.110.1	StaticSeries.StaticMember	455
2.111	ThreeDProperties	455
2.111.1	ThreeDProperties.Clustered	457
2.111.2	ThreeDProperties.DepthRatio	457
2.111.3	ThreeDProperties.DrawingStyle	457
2.111.4	ThreeDProperties.Enabled	458
2.111.5	ThreeDProperties.GapDepth	458
2.111.6	ThreeDProperties.HeightRatio	459
2.111.7	ThreeDProperties.Inclination	459
2.111.8	ThreeDProperties.Perspective	459
2.111.9	ThreeDProperties.ProjectionMode	460
2.111.10	ThreeDProperties.Rotation	460
2.111.11	ThreeDProperties.Shading	461
2.111.12	ThreeDProperties.WallThickness	461
2.112	Title	462
2.112.1	Title.Caption	462
2.112.2	Title.Position	463
2.112.3	Title.Style	463
2.113	ValueAxis	464
2.113.1	ValueAxis.Axis	464
2.114	ChartAnnotations	465
2.114.1	ChartAnnotations.ChartAnnotation	465
2.115	ChartAnnotation	465
2.116	ChartAreas	466
2.116.1	ChartAreas.ChartArea	466
2.117	ChartArea	467
2.117.1	ChartArea.Name	468
2.117.2	ChartArea.AlignOrientation	469
2.117.3	ChartArea.AlignWithChartArea	469
2.117.4	ChartArea.ChartAlignType	469
2.117.5	ChartArea.ChartCategoryAxes	470
2.117.6	ChartArea.ChartElementPosition	470
2.117.7	ChartArea.ChartInnerPlotPosition	471
2.117.8	ChartArea.ChartThreeDProperties	471
2.117.9	ChartArea.ChartValueAxes	471
2.117.10	ChartArea.EquallySizedAxesFont	472
2.117.11	ChartArea.Hidden	472
2.117.12	ChartArea.Style	472
2.118	ChartAlignType	473
2.118.1	ChartAlignType.AxesView	473
2.118.2	ChartAlignType.Cursor	474
2.118.3	ChartAlignType.InnerPlotPosition	474
2.118.4	ChartAlignType.Position	475
2.119	ChartCategoryAxes	475
2.119.1	ChartCategoryAxes.ChartAxis	476
2.120	ChartAxis	476
2.120.1	ChartAxis.Name	480
2.120.2	ChartAxis.AllowLabelRotation	480
2.120.3	ChartAxis.Angle	480
2.120.4	ChartAxis.Arrows	481
2.120.5	ChartAxis.ChartAxisScaleBreak	481
2.120.6	ChartAxis.ChartAxisTitle	482
2.120.7	ChartAxis.ChartMajorGridLines	482
2.120.8	ChartAxis.ChartMajorTickMarks	482
2.120.9	ChartAxis.ChartMinorGridLines	483
2.120.10	ChartAxis.ChartMinorTickMarks	483



2.120.11	ChartAxis.ChartStripLines	483
2.120.12	ChartAxis.CrossAt	483
2.120.13	ChartAxis.CustomProperties	484
2.120.14	ChartAxis.HideEndLabels	484
2.120.15	ChartAxis.HideLabels	485
2.120.16	ChartAxis.IncludeZero	485
2.120.17	ChartAxis.Interlaced	485
2.120.18	ChartAxis.InterlacedColor	486
2.120.19	ChartAxis.Interval	486
2.120.20	ChartAxis.IntervalOffset	486
2.120.21	ChartAxis.IntervalOffsetType	487
2.120.22	ChartAxis.IntervalType	487
2.120.23	ChartAxis.LabelInterval	488
2.120.24	ChartAxis.LabelIntervalOffset	489
2.120.25	ChartAxis.LabelIntervalOffsetType	489
2.120.26	ChartAxis.LabelIntervalType	490
2.120.27	ChartAxis.LabelsAutoFitDisabled	490
2.120.28	ChartAxis.Location	491
2.120.29	ChartAxis.LogBase	491
2.120.30	ChartAxis.LogScale	492
2.120.31	ChartAxis.Margin	492
2.120.32	ChartAxis.MarksAlwaysAtPlotEdge	493
2.120.33	ChartAxis.MaxFontSize	493
2.120.34	ChartAxis.Maximum	493
2.120.35	ChartAxis.MinFontSize	494
2.120.36	ChartAxis.Minimum	494
2.120.37	ChartAxis.OffsetLabels	494
2.120.38	ChartAxis.PreventFontGrow	495
2.120.39	ChartAxis.PreventFontShrink	495
2.120.40	ChartAxis.PreventLabelOffset	495
2.120.41	ChartAxis.PreventWordWrap	496
2.120.42	ChartAxis.Reverse	496
2.120.43	ChartAxis.Scalar	496
2.120.44	ChartAxis.Style	497
2.120.45	ChartAxis.VariableAutoInterval	497
2.120.46	ChartAxis.Visible	498
2.121	ChartAxisScaleBreak	498
2.121.1	ChartAxisScaleBreak.BreakLineType	499
2.121.2	ChartAxisScaleBreak.CollapsibleSpaceThreshold	500
2.121.3	ChartAxisScaleBreak.Enabled	500
2.121.4	ChartAxisScaleBreak.IncludeZero	501
2.121.5	ChartAxisScaleBreak.MaxNumberOfBreaks	501
2.121.6	ChartAxisScaleBreak.Spacing	501
2.121.7	ChartAxisScaleBreak.Style	502
2.122	ChartAxisTitle	502
2.122.1	ChartAxisTitle.Caption	503
2.122.2	ChartAxisTitle.Position	503
2.122.3	ChartAxisTitle.Style	504
2.122.4	ChartAxisTitle.TextOrientation	504
2.123	ChartGridLines	505
2.123.1	ChartGridLines.Enabled	506
2.123.2	ChartGridLines.Interval	506
2.123.3	ChartGridLines.IntervalOffset	507
2.123.4	ChartGridLines.IntervalOffsetType	507
2.123.5	ChartGridLines.IntervalType	508
2.123.6	ChartGridLines.Style	509
2.124	ChartStripLines	509
2.124.1	ChartStripLines.ChartStripLine	510

2.125	ChartStripLine	510
2.125.1	ChartStripLine.ActionInfo	511
2.125.2	ChartStripLine.Interval	512
2.125.3	ChartStripLine.IntervalOffset	512
2.125.4	ChartStripLine.IntervalOffsetType	513
2.125.5	ChartStripLine.IntervalType	513
2.125.6	ChartStripLine.StripWidth	514
2.125.7	ChartStripLine.StripWidthType	514
2.125.8	ChartStripLine.Style	515
2.125.9	ChartStripLine.TextOrientation	515
2.125.10	ChartStripLine.Title	516
2.125.11	ChartStripLine.TitleAngle	516
2.125.12	ChartStripLine.ToolTip	517
2.126	ChartTickMarks	517
2.126.1	ChartTickMarks.Enabled	518
2.126.2	ChartTickMarks.Interval	519
2.126.3	ChartTickMarks.IntervalOffset	519
2.126.4	ChartTickMarks.IntervalOffsetType	520
2.126.5	ChartTickMarks.IntervalType	520
2.126.6	ChartTickMarks.Length	521
2.126.7	ChartTickMarks.Style	521
2.126.8	ChartTickMarks.Type	522
2.127	ChartElementPosition	522
2.127.1	ChartElementPosition.Height	523
2.127.2	ChartElementPosition.Left	524
2.127.3	ChartElementPosition.Top	524
2.127.4	ChartElementPosition.Width	524
2.128	ChartThreeDProperties	525
2.128.1	ChartThreeDProperties.Clustered	526
2.128.2	ChartThreeDProperties.DepthRatio	526
2.128.3	ChartThreeDProperties.Enabled	527
2.128.4	ChartThreeDProperties.GapDepth	527
2.128.5	ChartThreeDProperties.Inclination	528
2.128.6	ChartThreeDProperties.Perspective	528
2.128.7	ChartThreeDProperties.ProjectionMode	528
2.128.8	ChartThreeDProperties.Rotation	529
2.128.9	ChartThreeDProperties.Shading	529
2.128.10	ChartThreeDProperties.WallThickness	530
2.129	ChartValueAxes	530
2.129.1	ChartValueAxes.ChartAxis	531
2.130	ChartBorderSkin	531
2.130.1	ChartBorderSkin.ChartBorderSkinType	532
2.130.2	ChartBorderSkin.Style	533
2.131	ChartCodeParameters	533
2.131.1	ChartCodeParameters.ChartCodeParameter	534
2.132	ChartCodeParameter	534
2.132.1	ChartCodeParameter.Name	535
2.132.2	ChartCodeParameter.Value	535
2.133	ChartCustomPaletteColors	535
2.133.1	ChartCustomPaletteColors.ChartCustomPaletteColor	536
2.134	ChartDerivedSeriesCollection	537
2.134.1	ChartDerivedSeriesCollection.ChartDerivedSeries	537
2.135	ChartDerivedSeries	538
2.135.1	ChartDerivedSeries.ChartFormulaParameters	540
2.135.2	ChartDerivedSeries.ChartSeries	540
2.135.3	ChartDerivedSeries.DerivedSeriesFormula	540
2.135.4	ChartDerivedSeries.SourceChartSeriesName	544
2.136	ChartFormulaParameters	544

2.136.1	ChartFormulaParameters.ChartFormulaParameter.....	545
2.137	ChartFormulaParameter.....	545
2.137.1	ChartFormulaParameter.Name.....	546
2.137.2	ChartFormulaParameter.Source.....	547
2.137.3	ChartFormulaParameter.Value.....	547
2.138	ChartSeriesCollection.....	547
2.138.1	ChartSeriesCollection.ChartSeries.....	548
2.139	ChartDataLabel.....	548
2.139.1	ChartDataLabel.ActionInfo.....	549
2.139.2	ChartDataLabel.Label.....	550
2.139.3	ChartDataLabel.Position.....	550
2.139.4	ChartDataLabel.Rotation.....	551
2.139.5	ChartDataLabel.Style.....	551
2.139.6	ChartDataLabel.ToolTip.....	552
2.139.7	ChartDataLabel.UseValueAsLabel.....	552
2.139.8	ChartDataLabel.Visible.....	552
2.140	ChartDataPoints.....	553
2.140.1	ChartDataPoints.ChartDataPoint.....	553
2.141	ChartDataPoint.....	554
2.141.1	ChartDataPoint.ActionInfo.....	556
2.141.2	ChartDataPoint.AxisLabel.....	556
2.141.3	ChartDataPoint.ChartDataLabel.....	556
2.141.4	ChartDataPoint.ChartDataPointValues.....	557
2.141.5	ChartDataPoint.ChartItemInLegend.....	557
2.141.6	ChartDataPoint.ChartMarker.....	557
2.141.7	ChartDataPoint.CustomProperties.....	558
2.141.8	ChartDataPoint.DataElementName.....	558
2.141.9	ChartDataPoint.DataElementOutput.....	558
2.141.10	ChartDataPoint.Style.....	559
2.141.11	ChartDataPoint.ToolTip.....	559
2.141.12	ChartDataPoint.DataSetName.....	560
2.141.13	ChartDataPoint.Relationships.....	560
2.142	ChartDataPointValues.....	560
2.142.1	ChartDataPointValues.End.....	562
2.142.2	ChartDataPointValues.High.....	562
2.142.3	ChartDataPointValues.Low.....	563
2.142.4	ChartDataPointValues.Mean.....	563
2.142.5	ChartDataPointValues.Median.....	564
2.142.6	ChartDataPointValues.Size.....	564
2.142.7	ChartDataPointValues.Start.....	564
2.142.8	ChartDataPointValues.X.....	565
2.142.9	ChartDataPointValues.Y.....	565
2.142.10	ChartDataPointValues.HighlightX.....	566
2.142.11	ChartDataPointValues.HighlightY.....	566
2.142.12	ChartDataPointValues.HighlightSize.....	566
2.142.13	ChartDataPointValues.FormatX.....	567
2.142.14	ChartDataPointValues.FormatY.....	567
2.142.15	ChartDataPointValues.FormatSize.....	568
2.142.16	ChartDataPointValues.CurrencyLanguageX.....	568
2.142.17	ChartDataPointValues.CurrencyLanguageY.....	568
2.142.18	ChartDataPointValues.CurrencyLanguageSize.....	569
2.143	ChartEmptyPoints.....	569
2.143.1	ChartEmptyPoints.ActionInfo.....	570
2.143.2	ChartEmptyPoints.AxisLabel.....	571
2.143.3	ChartEmptyPoints.ChartDataLabel.....	571
2.143.4	ChartEmptyPoints.ChartMarker.....	571
2.143.5	ChartEmptyPoints.CustomProperties.....	572
2.143.6	ChartEmptyPoints.Style.....	572

2.143.7	ChartEmptyPoints.ToolTip	572
2.144	ChartItemInLegend	573
2.144.1	ChartItemInLegend.ActionInfo	574
2.144.2	ChartItemInLegend.Hidden	574
2.144.3	ChartItemInLegend.LegendText	574
2.144.4	ChartItemInLegend.ToolTip	575
2.145	ChartMarker	575
2.145.1	ChartMarker.Size	576
2.145.2	ChartMarker.Style	576
2.145.3	ChartMarker.Type	577
2.146	ChartSmartLabel	577
2.146.1	ChartSmartLabel.AllowOutsidePlotArea	579
2.146.2	ChartSmartLabel.CalloutBackColor	579
2.146.3	ChartSmartLabel.CalloutLineAnchor	580
2.146.4	ChartSmartLabel.CalloutLineColor	580
2.146.5	ChartSmartLabel.CalloutLineStyle	581
2.146.6	ChartSmartLabel.CalloutLineWidth	581
2.146.7	ChartSmartLabel.CalloutStyle	582
2.146.8	ChartSmartLabel.ChartNoMoveDirections	582
2.146.9	ChartSmartLabel.Disabled	583
2.146.10	ChartSmartLabel.MarkerOverlapping	583
2.146.11	ChartSmartLabel.MaxMovingDistance	583
2.146.12	ChartSmartLabel.MinMovingDistance	584
2.146.13	ChartSmartLabel.ShowOverlapped	584
2.147	ChartNoMoveDirections	584
2.147.1	ChartNoMoveDirections.Down	586
2.147.2	ChartNoMoveDirections.DownLeft	586
2.147.3	ChartNoMoveDirections.DownRight	586
2.147.4	ChartNoMoveDirections.Left	587
2.147.5	ChartNoMoveDirections.Right	587
2.147.6	ChartNoMoveDirections.Up	587
2.147.7	ChartNoMoveDirections.UpLeft	588
2.147.8	ChartNoMoveDirections.UpRight	588
2.148	ChartHierarchy	588
2.148.1	ChartHierarchy.ChartMembers	589
2.148.2	ChartHierarchy.EnableDrilldown	590
2.149	ChartMembers	590
2.149.1	ChartMembers.ChartMember	591
2.150	ChartMember	591
2.150.1	ChartMember.ChartMembers	593
2.150.2	ChartMember.CustomProperties	593
2.150.3	ChartMember.DataElementName	593
2.150.4	ChartMember.DataElementOutput	594
2.150.5	ChartMember.Group	594
2.150.6	ChartMember.Label	595
2.150.7	ChartMember.SortExpressions	595
2.151	ChartLegends	596
2.151.1	ChartLegends.ChartLegend	596
2.152	ChartLegend	597
2.152.1	ChartLegend.Name	599
2.152.2	ChartLegend.AutoFitTextDisabled	599
2.152.3	ChartLegend.ChartElementPosition	599
2.152.4	ChartLegend.ChartLegendColumns	600
2.152.5	ChartLegend.ChartLegendTitle	600
2.152.6	ChartLegend.ColumnSeparator	600
2.152.7	ChartLegend.ColumnSeparatorColor	601
2.152.8	ChartLegend.ColumnSpacing	601
2.152.9	ChartLegend.DockOutsideChartArea	602

2.152.10	ChartLegend.DockToChartArea	602
2.152.11	ChartLegend.EquallySpacedItems	603
2.152.12	ChartLegend.HeaderSeparator	603
2.152.13	ChartLegend.HeaderSeparatorColor	604
2.152.14	ChartLegend.Hidden	604
2.152.15	ChartLegend.InterlacedRows	604
2.152.16	ChartLegend.InterlacedRowsColor	605
2.152.17	ChartLegend.Layout	605
2.152.18	ChartLegend.MaxAutoSize	606
2.152.19	ChartLegend.MinFontSize	606
2.152.20	ChartLegend.Position	606
2.152.21	ChartLegend.Reversed	607
2.152.22	ChartLegend.Style	608
2.152.23	ChartLegend.TextWrapThreshold	608
2.153	ChartLegendColumns	608
2.153.1	ChartLegendColumns.ChartLegendColumn	609
2.154	ChartLegendColumn	609
2.154.1	ChartLegendColumn.Name	611
2.154.2	ChartLegendColumn.ActionInfo	611
2.154.3	ChartLegendColumn.ColumnType	612
2.154.4	ChartLegendColumn.MaximumWidth	612
2.154.5	ChartLegendColumn.MinimumWidth	612
2.154.6	ChartLegendColumn.SeriesSymbolHeight	613
2.154.7	ChartLegendColumn.SeriesSymbolWidth	613
2.154.8	ChartLegendColumn.Style	613
2.154.9	ChartLegendColumn.ToolTip	614
2.154.10	ChartLegendColumn.Value	614
2.155	ChartLegendTitle	614
2.155.1	ChartLegendTitle.Caption	615
2.155.2	ChartLegendTitle.Style	615
2.155.3	ChartLegendTitle.TitleSeparator	616
2.156	ChartTitles	616
2.156.1	ChartTitles.ChartTitle	617
2.157	ChartTitle	617
2.157.1	ChartTitle.Name	619
2.157.2	ChartTitle.ActionInfo	619
2.157.3	ChartTitle.Caption	620
2.157.4	ChartTitle.ChartElementPosition	620
2.157.5	ChartTitle.DockOffset	620
2.157.6	ChartTitle.DockOutsideChartArea	621
2.157.7	ChartTitle.DockToChartArea	621
2.157.8	ChartTitle.Hidden	621
2.157.9	ChartTitle.Position	622
2.157.10	ChartTitle.Style	622
2.157.11	ChartTitle.TextOrientation	623
2.157.12	ChartTitle.ToolTip	623
2.158	ChartLegendColumnHeader	624
2.158.1	ChartLegendColumnHeader.Style	624
2.158.2	ChartLegendColumnHeader.Value	625
2.159	ChartLegendCustomItems	625
2.159.1	ChartLegendCustomItems.ChartLegendCustomItem	626
2.160	ChartLegendCustomItem	626
2.160.1	ChartLegendCustomItem.Name	627
2.160.2	ChartLegendCustomItem.ActionInfo	627
2.160.3	ChartLegendCustomItem.ChartLegendCustomItemCells	628
2.160.4	ChartLegendCustomItem.ChartMarker	628
2.160.5	ChartLegendCustomItem.Separator	628
2.160.6	ChartLegendCustomItem.SeparatorColor	629

2.160.7	ChartLegendCustomItem.Style	629
2.160.8	ChartLegendCustomItem.ToolTip	629
2.161	ChartLegendCustomItemCells	630
2.161.1	ChartLegendCustomItemCells.ChartLegendCustomItemCell	630
2.162	ChartLegendCustomItemCell	631
2.162.1	ChartLegendCustomItemCell.Name	633
2.162.2	ChartLegendCustomItemCell.ActionInfo	633
2.162.3	ChartLegendCustomItemCell.Alignment	633
2.162.4	ChartLegendCustomItemCell.BottomMargin	634
2.162.5	ChartLegendCustomItemCell.CellSpan	634
2.162.6	ChartLegendCustomItemCell.CellType	634
2.162.7	ChartLegendCustomItemCell.ImageHeight	635
2.162.8	ChartLegendCustomItemCell.ImageWidth	635
2.162.9	ChartLegendCustomItemCell.LeftMargin	635
2.162.10	ChartLegendCustomItemCell.RightMargin	636
2.162.11	ChartLegendCustomItemCell.Style	636
2.162.12	ChartLegendCustomItemCell.SymbolHeight	636
2.162.13	ChartLegendCustomItemCell.SymbolWidth	637
2.162.14	ChartLegendCustomItemCell.Text	637
2.162.15	ChartLegendCustomItemCell.ToolTip	637
2.162.16	ChartLegendCustomItemCell.TopMargin	638
2.163	GaugePanel	638
2.163.1	GaugePanel.Name	641
2.163.2	GaugePanel.Style	642
2.163.3	GaugePanel.ActionInfo	642
2.163.4	GaugePanel.Bookmark	642
2.163.5	GaugePanel.CustomProperties	643
2.163.6	GaugePanel.DataElementName	643
2.163.7	GaugePanel.DataElementOutput	643
2.163.8	GaugePanel.DocumentMapLabel	644
2.163.9	GaugePanel.Height	644
2.163.10	GaugePanel.Left	645
2.163.11	GaugePanel.RepeatWith	645
2.163.12	GaugePanel.ToolTip	645
2.163.13	GaugePanel.Top	646
2.163.14	GaugePanel.Visibility	646
2.163.15	GaugePanel.Width	646
2.163.16	GaugePanel.ZIndex	647
2.163.17	GaugePanel.DataSetName	647
2.163.18	GaugePanel.Filters	648
2.163.19	GaugePanel.NoRowsMessage	648
2.163.20	GaugePanel.PageBreak	648
2.163.21	GaugePanel.PageName	649
2.163.22	GaugePanel.SortExpressions	649
2.163.23	GaugePanel.AntiAliasing	649
2.163.24	GaugePanel.AutoLayout	650
2.163.25	GaugePanel.BackFrame	650
2.163.26	GaugePanel.GaugeImages	651
2.163.27	GaugePanel.GaugeLabels	651
2.163.28	GaugePanel.GaugeMember	651
2.163.29	GaugePanel.LinearGauges	652
2.163.30	GaugePanel.NumericIndicators	652
2.163.31	GaugePanel.RadialGauges	652
2.163.32	GaugePanel.ShadowIntensity	653
2.163.33	GaugePanel.StateIndicators	653
2.163.34	GaugePanel.TextAntiAliasingQuality	653
2.163.35	GaugePanel.TopImage	654
2.163.36	GaugePanel.Relationship	654

2.164	BackFrame	655
2.164.1	BackFrame.FrameBackground	656
2.164.2	BackFrame.FrameImage	656
2.164.3	BackFrame.FrameShape	656
2.164.4	BackFrame.FrameStyle	658
2.164.5	BackFrame.FrameWidth	659
2.164.6	BackFrame.GlassEffect	659
2.164.7	BackFrame.Style	660
2.165	FrameBackground	660
2.165.1	FrameBackground.Style	661
2.166	FrameImage	661
2.166.1	FrameImage.MIMEType	662
2.166.2	FrameImage.Source	662
2.166.3	FrameImage.TransparentColor	663
2.166.4	FrameImage.Value	663
2.166.5	FrameImage.ClipImage	664
2.166.6	FrameImage.HueColor	664
2.166.7	FrameImage.Transparency	665
2.167	GaugeImages	665
2.167.1	GaugeImages.GaugeImage	666
2.168	GaugeImage	666
2.168.1	GaugeImage.Name	668
2.168.2	GaugeImage.ActionInfo	668
2.168.3	GaugeImage.Height	668
2.168.4	GaugeImage.Hidden	669
2.168.5	GaugeImage.Left	669
2.168.6	GaugeImage.ParentItem	669
2.168.7	GaugeImage.ToolTip	670
2.168.8	GaugeImage.Top	670
2.168.9	GaugeImage.Width	670
2.168.10	GaugeImage.ZIndex	671
2.168.11	GaugeImage.Angle	671
2.168.12	GaugeImage.MIMEType	671
2.168.13	GaugeImage.ResizeMode	672
2.168.14	GaugeImage.Source	672
2.168.15	GaugeImage.Transparency	673
2.168.16	GaugeImage.TransparentColor	673
2.168.17	GaugeImage.Value	673
2.169	GaugeLabels	674
2.169.1	GaugeLabels.GaugeLabel	674
2.170	GaugeLabel	675
2.170.1	GaugeLabel.Name	676
2.170.2	GaugeLabel.ActionInfo	677
2.170.3	GaugeLabel.Height	677
2.170.4	GaugeLabel.Hidden	677
2.170.5	GaugeLabel.Left	678
2.170.6	GaugeLabel.ParentItem	678
2.170.7	GaugeLabel.ToolTip	679
2.170.8	GaugeLabel.Top	679
2.170.9	GaugeLabel.Width	679
2.170.10	GaugeLabel.ZIndex	680
2.170.11	GaugeLabel.Angle	680
2.170.12	GaugeLabel.ResizeMode	681
2.170.13	GaugeLabel.Style	681
2.170.14	GaugeLabel.Text	681
2.170.15	GaugeLabel.TextShadowOffset	682
2.170.16	GaugeLabel.UseFontPercent	682
2.171	GaugeMember	682

2.171.1	GaugeMember.GaugeMember .....	683
2.171.2	GaugeMember.Group .....	684
2.171.3	GaugeMember.SortExpressions .....	684
2.172	LinearGauges .....	684
2.172.1	LinearGauges.LinearGauge .....	685
2.173	LinearGauge .....	685
2.173.1	LinearGauge.Name .....	687
2.173.2	LinearGauge.ActionInfo .....	687
2.173.3	LinearGauge.Height .....	688
2.173.4	LinearGauge.Hidden .....	688
2.173.5	LinearGauge.Left .....	689
2.173.6	LinearGauge.ParentItem .....	689
2.173.7	LinearGauge.ToolTip .....	689
2.173.8	LinearGauge.Top .....	690
2.173.9	LinearGauge.Width .....	690
2.173.10	LinearGauge.ZIndex .....	691
2.173.11	LinearGauge.AspectRatio .....	691
2.173.12	LinearGauge.BackFrame .....	691
2.173.13	LinearGauge.ClipContent .....	692
2.173.14	LinearGauge.GaugeScales .....	692
2.173.15	LinearGauge.TopImage .....	692
2.173.16	LinearGauge.Orientation .....	693
2.174	LinearScales .....	693
2.174.1	LinearScales.LinearScale .....	694
2.175	LinearScale .....	694
2.175.1	LinearScale.Name .....	697
2.175.2	LinearScale.ActionInfo .....	697
2.175.3	LinearScale.CustomLabels .....	697
2.175.4	LinearScale.GaugeMajorTickMarks .....	698
2.175.5	LinearScale.GaugeMinorTickMarks .....	698
2.175.6	LinearScale.GaugePointers .....	698
2.175.7	LinearScale.Hidden .....	699
2.175.8	LinearScale.Interval .....	699
2.175.9	LinearScale.IntervalOffset .....	699
2.175.10	LinearScale.Logarithmic .....	700
2.175.11	LinearScale.LogarithmicBase .....	700
2.175.12	LinearScale.MaximumPin .....	701
2.175.13	LinearScale.MaximumValue .....	701
2.175.14	LinearScale.MinimumPin .....	701
2.175.15	LinearScale.MinimumValue .....	702
2.175.16	LinearScale.Multiplier .....	702
2.175.17	LinearScale.Reversed .....	703
2.175.18	LinearScale.ScaleLabels .....	703
2.175.19	LinearScale.ScaleRanges .....	703
2.175.20	LinearScale.Style .....	704
2.175.21	LinearScale.TickMarksOnTop .....	704
2.175.22	LinearScale.ToolTip .....	704
2.175.23	LinearScale.Width .....	705
2.175.24	LinearScale.EndMargin .....	705
2.175.25	LinearScale.Position .....	705
2.175.26	LinearScale.StartMargin .....	706
2.176	CustomLabels .....	706
2.176.1	CustomLabels.CustomLabel .....	707
2.177	CustomLabel .....	707
2.177.1	CustomLabel.Name .....	709
2.177.2	CustomLabel.AllowUpsideDown .....	709
2.177.3	CustomLabel.DistanceFromScale .....	709
2.177.4	CustomLabel.FontAngle .....	710



2.177.5	CustomLabel.Hidden	710
2.177.6	CustomLabel.Placement	711
2.177.7	CustomLabel.RotateLabel	711
2.177.8	CustomLabel.Style	711
2.177.9	CustomLabel.Text	712
2.177.10	CustomLabel.TickMarkStyle	712
2.177.11	CustomLabel.UseFontPercent	712
2.177.12	CustomLabel.Value	713
2.178	TickMarkStyle	713
2.178.1	TickMarkStyle.DistanceFromScale	714
2.178.2	TickMarkStyle.EnableGradient	715
2.178.3	TickMarkStyle.GradientDensity	715
2.178.4	TickMarkStyle.Hidden	716
2.178.5	TickMarkStyle.Length	716
2.178.6	TickMarkStyle.Placement	716
2.178.7	TickMarkStyle.Shape	717
2.178.8	TickMarkStyle.Style	718
2.178.9	TickMarkStyle.TickMarkImage	718
2.178.10	TickMarkStyle.Width	718
2.179	GaugeInputValue	719
2.179.1	GaugeInputValue.AddConstant	720
2.179.2	GaugeInputValue.DataElementName	721
2.179.3	GaugeInputValue.DataElementOutput	721
2.179.4	GaugeInputValue.Formula	722
2.179.5	GaugeInputValue.MaxPercent	722
2.179.6	GaugeInputValue.MinPercent	723
2.179.7	GaugeInputValue.Multiplier	723
2.179.8	GaugeInputValue.Value	724
2.180	GaugeTickMarks	724
2.180.1	GaugeTickMarks.Interval	725
2.180.2	GaugeTickMarks.IntervalOffset	726
2.180.3	GaugeTickMarks.DistanceFromScale	726
2.180.4	GaugeTickMarks.EnableGradient	727
2.180.5	GaugeTickMarks.GradientDensity	727
2.180.6	GaugeTickMarks.Hidden	727
2.180.7	GaugeTickMarks.Length	728
2.180.8	GaugeTickMarks.Placement	728
2.180.9	GaugeTickMarks.Shape	729
2.180.10	GaugeTickMarks.Style	729
2.180.11	GaugeTickMarks.TickMarkImage	730
2.180.12	GaugeTickMarks.Width	730
2.181	LinearPointers	730
2.181.1	LinearPointers.LinearPointer	731
2.182	LinearPointer	732
2.182.1	LinearPointer.Name	733
2.182.2	LinearPointer.ActionInfo	734
2.182.3	LinearPointer.BarStart	734
2.182.4	LinearPointer.DistanceFromScale	735
2.182.5	LinearPointer.GaugeInputValue	735
2.182.6	LinearPointer.Hidden	735
2.182.7	LinearPointer.MarkerLength	736
2.182.8	LinearPointer.MarkerStyle	736
2.182.9	LinearPointer.Placement	737
2.182.10	LinearPointer.PointerImage	737
2.182.11	LinearPointer.SnappingEnabled	737
2.182.12	LinearPointer.SnappingInterval	738
2.182.13	LinearPointer.Style	738
2.182.14	LinearPointer.ToolTip	739

2.182.15	LinearPointer.Width	739
2.182.16	LinearPointer.Thermometer	739
2.182.17	LinearPointer.Type	740
2.183	PointerImage	740
2.183.1	PointerImage.MIMETYPE	741
2.183.2	PointerImage.Source	742
2.183.3	PointerImage.TransparentColor	742
2.183.4	PointerImage.Value	742
2.183.5	PointerImage.HueColor	743
2.183.6	PointerImage.OffsetX	743
2.183.7	PointerImage.OffsetY	744
2.183.8	PointerImage.Transparency	744
2.184	Thermometer	744
2.184.1	Thermometer.BulbOffset	745
2.184.2	Thermometer.BulbSize	746
2.184.3	Thermometer.Style	746
2.184.4	Thermometer.ThermometerStyle	746
2.185	ScaleLabels	747
2.185.1	ScaleLabels.AllowUpsideDown	748
2.185.2	ScaleLabels.DistanceFromScale	749
2.185.3	ScaleLabels.FontAngle	749
2.185.4	ScaleLabels.Hidden	749
2.185.5	ScaleLabels.Interval	750
2.185.6	ScaleLabels.IntervalOffset	750
2.185.7	ScaleLabels.Placement	751
2.185.8	ScaleLabels.RotateLabels	751
2.185.9	ScaleLabels.ShowEndLabels	751
2.185.10	ScaleLabels.Style	752
2.185.11	ScaleLabels.UseFontPercent	752
2.186	ScalePin	753
2.186.1	ScalePin.DistanceFromScale	754
2.186.2	ScalePin.EnableGradient	755
2.186.3	ScalePin.GradientDensity	755
2.186.4	ScalePin.Hidden	755
2.186.5	ScalePin.Length	756
2.186.6	ScalePin.Placement	756
2.186.7	ScalePin.Shape	757
2.186.8	ScalePin.Style	757
2.186.9	ScalePin.TickMarkImage	758
2.186.10	ScalePin.Width	758
2.186.11	ScalePin.Enable	758
2.186.12	ScalePin.Location	759
2.186.13	ScalePin.PinLabel	759
2.187	PinLabel	759
2.187.1	PinLabel.AllowUpsideDown	761
2.187.2	PinLabel.DistanceFromScale	761
2.187.3	PinLabel.FontAngle	761
2.187.4	PinLabel.Placement	762
2.187.5	PinLabel.RotateLabel	762
2.187.6	PinLabel.Style	763
2.187.7	PinLabel.Text	763
2.187.8	PinLabel.UseFontPercent	763
2.188	ScaleRanges	764
2.188.1	ScaleRanges.ScaleRange	765
2.189	ScaleRange	765
2.189.1	ScaleRange.Name	767
2.189.2	ScaleRange.ActionInfo	767
2.189.3	ScaleRange.BackgroundGradientType	767

2.189.4	ScaleRange.DistanceFromScale	768
2.189.5	ScaleRange.EndValue	768
2.189.6	ScaleRange.EndWidth	769
2.189.7	ScaleRange.Hidden	769
2.189.8	ScaleRange.InRangeBarPointerColor	769
2.189.9	ScaleRange.InRangeLabelColor	770
2.189.10	ScaleRange.InRangeTickMarksColor	770
2.189.11	ScaleRange.Placement	770
2.189.12	ScaleRange.StartValue	771
2.189.13	ScaleRange.StartWidth	771
2.189.14	ScaleRange.Style	772
2.189.15	ScaleRange.ToolTip	772
2.190	TopImage	772
2.190.1	TopImage.MIMETYPE	774
2.190.2	TopImage.Source	774
2.190.3	TopImage.TransparentColor	774
2.190.4	TopImage.Value	775
2.190.5	TopImage.HueColor	775
2.191	NumericIndicators	776
2.191.1	NumericIndicators.NumericIndicator	776
2.192	NumericIndicator	777
2.192.1	NumericIndicator.Name	779
2.192.2	NumericIndicator.ActionInfo	780
2.192.3	NumericIndicator.Height	780
2.192.4	NumericIndicator.Hidden	780
2.192.5	NumericIndicator.Left	781
2.192.6	NumericIndicator.ParentItem	781
2.192.7	NumericIndicator.ToolTip	781
2.192.8	NumericIndicator.Top	782
2.192.9	NumericIndicator.Width	782
2.192.10	NumericIndicator.ZIndex	782
2.192.11	NumericIndicator.DecimalDigitColor	783
2.192.12	NumericIndicator.DecimalDigits	783
2.192.13	NumericIndicator.DigitColor	783
2.192.14	NumericIndicator.Digits	784
2.192.15	NumericIndicator.GaugeInputValue	784
2.192.16	NumericIndicator.IndicatorStyle	784
2.192.17	NumericIndicator.LedDimColor	785
2.192.18	NumericIndicator.MaximumValue	785
2.192.19	NumericIndicator.MinimumValue	786
2.192.20	NumericIndicator.Multiplier	786
2.192.21	NumericIndicator.NumericIndicatorRanges	786
2.192.22	NumericIndicator.Offstring	787
2.192.23	NumericIndicator.OutOfRangeString	787
2.192.24	NumericIndicator.ResizeMode	787
2.192.25	NumericIndicator.SeparatorColor	787
2.192.26	NumericIndicator.SeparatorWidth	788
2.192.27	NumericIndicator.ShowDecimalPoint	788
2.192.28	NumericIndicator.ShowLeadingZeros	788
2.192.29	NumericIndicator.ShowSign	789
2.192.30	NumericIndicator.SnappingEnabled	789
2.192.31	NumericIndicator.SnappingInterval	790
2.192.32	NumericIndicator.Style	790
2.192.33	NumericIndicator.UseFontPercent	790
2.193	NumericIndicatorRanges	791
2.193.1	NumericIndicatorRanges.NumericIndicatorRange	791
2.194	NumericIndicatorRange	792
2.194.1	NumericIndicatorRange.Name	793

2.194.2	NumericIndicatorRange.StartValue	793
2.194.3	NumericIndicatorRange.EndValue	793
2.194.4	NumericIndicatorRange.DecimalDigitColor	794
2.194.5	NumericIndicatorRange.DigitColor	794
2.195	RadialGauges	794
2.195.1	RadialGauges.RadialGauge	795
2.196	RadialGauge	795
2.196.1	RadialGauge.Name	797
2.196.2	RadialGauge.ActionInfo	798
2.196.3	RadialGauge.Height	798
2.196.4	RadialGauge.Hidden	798
2.196.5	RadialGauge.Left	799
2.196.6	RadialGauge.ParentItem	799
2.196.7	RadialGauge.ToolTip	799
2.196.8	RadialGauge.Top	800
2.196.9	RadialGauge.Width	800
2.196.10	RadialGauge.ZIndex	801
2.196.11	RadialGauge.AspectRatio	801
2.196.12	RadialGauge.BackFrame	801
2.196.13	RadialGauge.ClipContent	802
2.196.14	RadialGauge.GaugeScales	802
2.196.15	RadialGauge.TopImage	802
2.196.16	RadialGauge.PivotX	803
2.196.17	RadialGauge.PivotY	803
2.197	RadialScales	803
2.197.1	RadialScales.RadialScale	804
2.198	RadialScale	804
2.198.1	RadialScale.Name	807
2.198.2	RadialScale.ActionInfo	807
2.198.3	RadialScale.CustomLabels	808
2.198.4	RadialScale.GaugeMajorTickMarks	808
2.198.5	RadialScale.GaugeMinorTickMarks	808
2.198.6	RadialScale.GaugePointers	809
2.198.7	RadialScale.Hidden	809
2.198.8	RadialScale.Interval	809
2.198.9	RadialScale.IntervalOffset	810
2.198.10	RadialScale.Logarithmic	810
2.198.11	RadialScale.LogarithmicBase	810
2.198.12	RadialScale.MaximumPin	811
2.198.13	RadialScale.MaximumValue	811
2.198.14	RadialScale.MinimumPin	812
2.198.15	RadialScale.MinimumValue	812
2.198.16	RadialScale.Multiplier	812
2.198.17	RadialScale.Reversed	813
2.198.18	RadialScale.ScaleLabels	813
2.198.19	RadialScale.ScaleRanges	813
2.198.20	RadialScale.Style	814
2.198.21	RadialScale.TickMarksOnTop	814
2.198.22	RadialScale.ToolTip	814
2.198.23	RadialScale.Width	815
2.198.24	RadialScale.Radius	815
2.198.25	RadialScale.StartAngle	816
2.198.26	RadialScale.SweepAngle	816
2.199	RadialPointers	816
2.199.1	RadialPointers.RadialPointer	817
2.200	RadialPointer	817
2.200.1	RadialPointer.Name	819
2.200.2	RadialPointer.ActionInfo	819

2.200.3	RadialPointer.BarStart	820
2.200.4	RadialPointer.DistanceFromScale	820
2.200.5	RadialPointer.GaugeInputValue	821
2.200.6	RadialPointer.Hidden	821
2.200.7	RadialPointer.MarkerLength	821
2.200.8	RadialPointer.MarkerStyle	822
2.200.9	RadialPointer.Placement	822
2.200.10	RadialPointer.PointerImage	823
2.200.11	RadialPointer.SnappingEnabled	823
2.200.12	RadialPointer.SnappingInterval	823
2.200.13	RadialPointer.Style	824
2.200.14	RadialPointer.ToolTip	824
2.200.15	RadialPointer.Width	825
2.200.16	RadialPointer.NeedleStyle	825
2.200.17	RadialPointer.PointerCap	826
2.200.18	RadialPointer.Type	826
2.201	PointerCap	827
2.201.1	PointerCap.CapImage	828
2.201.2	PointerCap.CapStyle	828
2.201.3	PointerCap.Hidden	829
2.201.4	PointerCap.OnTop	829
2.201.5	PointerCap.Reflection	829
2.201.6	PointerCap.Style	830
2.201.7	PointerCap.Width	830
2.202	CapImage	830
2.202.1	CapImage.MIMETYPE	832
2.202.2	CapImage.Source	832
2.202.3	CapImage.TransparentColor	832
2.202.4	CapImage.Value	833
2.202.5	CapImage.HueColor	833
2.202.6	CapImage.OffsetX	834
2.202.7	CapImage.OffsetY	834
2.203	StateIndicators	834
2.203.1	StateIndicators.StateIndicator	835
2.204	StateIndicator	835
2.204.1	StateIndicator.Name	838
2.204.2	StateIndicator.ActionInfo	838
2.204.3	StateIndicator.Height	838
2.204.4	StateIndicator.Hidden	839
2.204.5	StateIndicator.Left	839
2.204.6	StateIndicator.MaximumValue	840
2.204.7	StateIndicator.MinimumValue	840
2.204.8	StateIndicator.ParentItem	840
2.204.9	StateIndicator.ScaleFactor	841
2.204.10	StateIndicator.StateDataElementName	841
2.204.11	StateIndicator.StateDataElementOutput	841
2.204.12	StateIndicator.ToolTip	842
2.204.13	StateIndicator.Top	842
2.204.14	StateIndicator.TransformationScope	843
2.204.15	StateIndicator.TransformationType	843
2.204.16	StateIndicator.Width	844
2.204.17	StateIndicator.ZIndex	844
2.204.18	StateIndicator.Angle	845
2.204.19	StateIndicator.GaugeInputValue	845
2.204.20	StateIndicator.IndicatorStates	845
2.204.21	StateIndicator.IndicatorStyle	846
2.204.22	StateIndicator.ResizeMode	848
2.204.23	StateIndicator.IndicatorImage	848

2.204.24	StateIndicator.StateImage	848
2.204.25	StateIndicator.Style	849
2.204.26	StateIndicator.Text	849
2.204.27	StateIndicator.UseFontPercent	849
2.205	IndicatorStates	850
2.205.1	IndicatorStates.IndicatorState	850
2.206	IndicatorState	851
2.206.1	IndicatorState.Name	852
2.206.2	IndicatorState.Color	852
2.206.3	IndicatorState.IndicatorStyle	853
2.206.4	IndicatorState.Style	855
2.206.5	IndicatorState.StartValue	855
2.206.6	IndicatorState.EndValue	855
2.206.7	IndicatorState.IndicatorImage	856
2.206.8	IndicatorState.ScaleFactor	856
2.206.9	IndicatorState.StateImage	856
2.206.10	IndicatorState.Text	857
2.207	StateImage	857
2.207.1	StateImage.MIMETYPE	858
2.207.2	StateImage.Source	858
2.207.3	StateImage.TransparentColor	858
2.207.4	StateImage.Value	859
2.207.5	StateImage.HueColor	859
2.207.6	StateImage.Transparency	859
2.208	IndicatorImage	859
2.208.1	IndicatorImage.MIMETYPE	860
2.208.2	IndicatorImage.Source	861
2.208.3	IndicatorImage.TransparentColor	861
2.208.4	IndicatorImage.Value	861
2.208.5	IndicatorImage.HueColor	862
2.208.6	IndicatorImage.Transparency	862
2.209	Map	863
2.209.1	Map.Name	865
2.209.2	Map.ActionInfo	865
2.209.3	Map.AntiAliasing	866
2.209.4	Map.Bookmark	866
2.209.5	Map.CustomProperties	866
2.209.6	Map.DataElementName	867
2.209.7	Map.DataElementOutput	867
2.209.8	Map.DocumentMapLabel	868
2.209.9	Map.Height	868
2.209.10	Map.Left	868
2.209.11	Map.MapBorderSkin	869
2.209.12	Map.MapColorScale	869
2.209.13	Map.MapDataRegions	869
2.209.14	Map.MapDistanceScale	870
2.209.15	Map.MapLayers	870
2.209.16	Map.MapLegends	870
2.209.17	Map.MapTitles	871
2.209.18	Map.MapViewport	871
2.209.19	Map.MaximumSpatialElementCount	871
2.209.20	Map.MaximumTotalPointCount	872
2.209.21	Map.PageBreak	872
2.209.22	Map.PageName	872
2.209.23	Map.RepeatWith	873
2.209.24	Map.ShadowIntensity	873
2.209.25	Map.Style	873
2.209.26	Map.TextAntiAliasingQuality	874

2.209.27	Map.TileLanguage	874
2.209.28	Map.ToolTip	875
2.209.29	Map.Top	875
2.209.30	Map.Visibility	876
2.209.31	Map.Width	876
2.209.32	Map.ZIndex	876
2.210	MapBorderSkin	877
2.210.1	MapBorderSkin.MapBorderSkinType	877
2.210.2	MapBorderSkin.Style	878
2.211	MapColorScale	878
2.211.1	MapColorScale.ActionInfo	880
2.211.2	MapColorScale.BottomMargin	880
2.211.3	MapColorScale.ColorBarBorderColor	881
2.211.4	MapColorScale.DockOutsideViewport	881
2.211.5	MapColorScale.Hidden	881
2.211.6	MapColorScale.HideEndLabels	882
2.211.7	MapColorScale.LabelBehavior	882
2.211.8	MapColorScale.LabelFormat	883
2.211.9	MapColorScale.LabelInterval	883
2.211.10	MapColorScale.LabelPlacement	883
2.211.11	MapColorScale.LeftMargin	884
2.211.12	MapColorScale.MapColorScaleTitle	884
2.211.13	MapColorScale.MapLocation	885
2.211.14	MapColorScale.MapSize	885
2.211.15	MapColorScale.NoDataText	885
2.211.16	MapColorScale.Position	886
2.211.17	MapColorScale.RangeGapColor	886
2.211.18	MapColorScale.RightMargin	887
2.211.19	MapColorScale.Style	887
2.211.20	MapColorScale.TickMarkLength	887
2.211.21	MapColorScale.ToolTip	888
2.211.22	MapColorScale.TopMargin	888
2.211.23	MapColorScale.ZIndex	888
2.212	MapColorScaleTitle	889
2.212.1	MapColorScaleTitle.Caption	889
2.212.2	MapColorScaleTitle.Style	890
2.213	MapLocation	890
2.213.1	MapLocation.Left	891
2.213.2	MapLocation.Top	891
2.213.3	MapLocation.Unit	892
2.214	MapSize	892
2.214.1	MapSize.Height	893
2.214.2	MapSize.Unit	893
2.214.3	MapSize.Width	894
2.215	MapDataRegions	894
2.215.1	MapDataRegions.MapDataRegion	895
2.216	MapDataRegion	895
2.216.1	MapDataRegion.Name	896
2.216.2	MapDataRegion.DataSetName	896
2.216.3	MapDataRegion.Filters	897
2.216.4	MapDataRegion.MapMember	897
2.216.5	MapDataRegion.Relationship	898
2.217	MapMember	898
2.217.1	MapMember.Group	899
2.217.2	MapMember.MapMember	899
2.218	MapDistanceScale	899
2.218.1	MapDistanceScale.ActionInfo	900
2.218.2	MapDistanceScale.BottomMargin	901

2.218.3	MapDistanceScale.DockOutsideViewport	901
2.218.4	MapDistanceScale.Hidden	902
2.218.5	MapDistanceScale.LeftMargin	902
2.218.6	MapDistanceScale.MapLocation	902
2.218.7	MapDistanceScale.MapSize	903
2.218.8	MapDistanceScale.Position	903
2.218.9	MapDistanceScale.RightMargin	904
2.218.10	MapDistanceScale.ScaleBorderColor	904
2.218.11	MapDistanceScale.ScaleColor	904
2.218.12	MapDistanceScale.Style	905
2.218.13	MapDistanceScale.ToolTip	905
2.218.14	MapDistanceScale.TopMargin	906
2.218.15	MapDistanceScale.ZIndex	906
2.219	MapLayers	906
2.219.1	MapLayers.MapLineLayer	907
2.219.2	MapLayers.MapPointLayer	907
2.219.3	MapLayers.MapPolygonLayer	908
2.219.4	MapLayers.MapTileLayer	908
2.220	MapLineLayer	908
2.220.1	MapLineLayer.Name	910
2.220.2	MapLineLayer.DataElementName	910
2.220.3	MapLineLayer.DataElementOutput	910
2.220.4	MapLineLayer.MapBindingFieldPairs	911
2.220.5	MapLineLayer.MapDataRegionName	911
2.220.6	MapLineLayer.MapFieldDefinitions	912
2.220.7	MapLineLayer.MapLineRules	912
2.220.8	MapLineLayer.MapLines	913
2.220.9	MapLineLayer.MapLineTemplate	913
2.220.10	MapLineLayer.MapSpatialDataRegion	913
2.220.11	MapLineLayer.MapShapefile	914
2.220.12	MapLineLayer.MapSpatialDataSet	914
2.220.13	MapLineLayer.MaximumZoom	915
2.220.14	MapLineLayer.MinimumZoom	915
2.220.15	MapLineLayer.Transparency	915
2.220.16	MapLineLayer.VisibilityMode	916
2.221	MapBindingFieldPairs	916
2.221.1	MapBindingFieldPairs.MapBindingFieldPair	917
2.222	MapBindingFieldPair	917
2.222.1	MapBindingFieldPair.BindingExpression	918
2.222.2	MapBindingFieldPair.FieldName	918
2.223	MapFieldDefinitions	919
2.223.1	MapFieldDefinitions.MapFieldDefinition	919
2.224	MapFieldDefinition	920
2.224.1	MapFieldDefinition.DataType	921
2.224.2	MapFieldDefinition.Name	921
2.225	MapLineRules	922
2.225.1	MapLineRules.MapColorPaletteRule	922
2.225.2	MapLineRules.MapColorRangeRule	923
2.225.3	MapLineRules.MapCustomColorRule	923
2.225.4	MapLineRules.MapSizeRule	924
2.226	MapColorPaletteRule	924
2.226.1	MapColorPaletteRule.BucketCount	925
2.226.2	MapColorPaletteRule.DataElementName	926
2.226.3	MapColorPaletteRule.DataElementOutput	926
2.226.4	MapColorPaletteRule.DataValue	927
2.226.5	MapColorPaletteRule.DistributionType	927
2.226.6	MapColorPaletteRule.EndValue	928
2.226.7	MapColorPaletteRule.LegendName	928



2.226.8	MapColorPaletteRule.LegendText	929
2.226.9	MapColorPaletteRule.MapBuckets	929
2.226.10	MapColorPaletteRule.Palette	929
2.226.11	MapColorPaletteRule.ShowInColorScale	930
2.226.12	MapColorPaletteRule.StartValue	930
2.227	MapBuckets	931
2.227.1	MapBuckets.MapBucket	931
2.228	MapBucket	932
2.228.1	MapBucket.EndValue	932
2.228.2	MapBucket.StartValue	933
2.229	MapColorRangeRule	933
2.229.1	MapColorRangeRule.BucketCount	934
2.229.2	MapColorRangeRule.DataElementName	935
2.229.3	MapColorRangeRule.DataElementOutput	935
2.229.4	MapColorRangeRule.DataValue	936
2.229.5	MapColorRangeRule.DistributionType	936
2.229.6	MapColorRangeRule.EndColor	937
2.229.7	MapColorRangeRule.EndValue	937
2.229.8	MapColorRangeRule.LegendName	938
2.229.9	MapColorRangeRule.LegendText	938
2.229.10	MapColorRangeRule.MapBuckets	938
2.229.11	MapColorRangeRule.MiddleColor	939
2.229.12	MapColorRangeRule.ShowInColorScale	939
2.229.13	MapColorRangeRule.StartColor	939
2.229.14	MapColorRangeRule.StartValue	940
2.230	MapCustomColorRule	940
2.230.1	MapCustomColorRule.BucketCount	941
2.230.2	MapCustomColorRule.DataElementName	942
2.230.3	MapCustomColorRule.DataElementOutput	942
2.230.4	MapCustomColorRule.DataValue	943
2.230.5	MapCustomColorRule.DistributionType	943
2.230.6	MapCustomColorRule.EndValue	944
2.230.7	MapCustomColorRule.LegendName	944
2.230.8	MapCustomColorRule.LegendText	945
2.230.9	MapCustomColorRule.MapBuckets	945
2.230.10	MapCustomColorRule.MapCustomColors	945
2.230.11	MapCustomColorRule.ShowInColorScale	946
2.230.12	MapCustomColorRule.StartValue	946
2.231	MapCustomColors	947
2.231.1	MapCustomColors.MapCustomColor	947
2.232	MapSizeRule	947
2.232.1	MapSizeRule.BucketCount	949
2.232.2	MapSizeRule.DataElementName	949
2.232.3	MapSizeRule.DataElementOutput	949
2.232.4	MapSizeRule.DataValue	950
2.232.5	MapSizeRule.DistributionType	950
2.232.6	MapSizeRule.EndSize	951
2.232.7	MapSizeRule.EndValue	951
2.232.8	MapSizeRule.LegendName	952
2.232.9	MapSizeRule.LegendText	952
2.232.10	MapSizeRule.MapBuckets	952
2.232.11	MapSizeRule.StartSize	953
2.232.12	MapSizeRule.StartValue	953
2.233	MapLines	953
2.233.1	MapLines.MapLine	954
2.234	MapLine	954
2.234.1	MapLine.MapFields	955
2.234.2	MapLine.MapLineTemplate	955

2.234.3	MapLine.UseCustomLineTemplate	956
2.234.4	MapLine.VectorData	956
2.235	MapFields	957
2.235.1	MapFields.MapField	957
2.236	MapField	957
2.236.1	MapField.Name	958
2.236.2	MapField.Value	958
2.237	MapLineTemplate	959
2.237.1	MapLineTemplate.ActionInfo	960
2.237.2	MapLineTemplate.DataElementLabel	960
2.237.3	MapLineTemplate.DataElementName	961
2.237.4	MapLineTemplate.DataElementOutput	961
2.237.5	MapLineTemplate.Hidden	961
2.237.6	MapLineTemplate.Label	962
2.237.7	MapLineTemplate.LabelPlacement	962
2.237.8	MapLineTemplate.OffsetX	963
2.237.9	MapLineTemplate.OffsetY	963
2.237.10	MapLineTemplate.Style	963
2.237.11	MapLineTemplate.ToolTip	964
2.237.12	MapLineTemplate.Width	964
2.238	MapShapefile	964
2.238.1	MapShapefile.MapFieldNames	965
2.238.2	MapShapefile.Source	965
2.239	MapFieldNames	966
2.239.1	MapFieldNames.MapFieldName	966
2.240	MapSpatialDataRegion	967
2.240.1	MapSpatialDataRegion.VectorData	967
2.241	MapSpatialDataSet	967
2.241.1	MapSpatialDataSet.DataSetName	968
2.241.2	MapSpatialDataSet.MapFieldNames	968
2.241.3	MapSpatialDataSet.SpatialField	969
2.242	MapPointLayer	969
2.242.1	MapPointLayer.Name	971
2.242.2	MapPointLayer.DataElementName	971
2.242.3	MapPointLayer.DataElementOutput	971
2.242.4	MapPointLayer.MapBindingFieldPairs	972
2.242.5	MapPointLayer.MapDataRegionName	972
2.242.6	MapPointLayer.MapFieldDefinitions	973
2.242.7	MapPointLayer.MapPointRules	973
2.242.8	MapPointLayer.MapPoints	973
2.242.9	MapPointLayer.MapMarkerTemplate	974
2.242.10	MapPointLayer.MapSpatialDataRegion	974
2.242.11	MapPointLayer.MapShapefile	975
2.242.12	MapPointLayer.MapSpatialDataSet	975
2.242.13	MapPointLayer.MaximumZoom	975
2.242.14	MapPointLayer.MinimumZoom	976
2.242.15	MapPointLayer.Transparency	976
2.242.16	MapPointLayer.VisibilityMode	977
2.243	MapMarkerTemplate	977
2.243.1	MapMarkerTemplate.ActionInfo	978
2.243.2	MapMarkerTemplate.DataElementLabel	979
2.243.3	MapMarkerTemplate.DataElementName	979
2.243.4	MapMarkerTemplate.DataElementOutput	979
2.243.5	MapMarkerTemplate.Hidden	980
2.243.6	MapMarkerTemplate.Label	980
2.243.7	MapMarkerTemplate.LabelPlacement	981
2.243.8	MapMarkerTemplate.MapMarker	981
2.243.9	MapMarkerTemplate.OffsetX	982

2.243.10	MapMarkerTemplate.OffsetY	982
2.243.11	MapMarkerTemplate.Size	982
2.243.12	MapMarkerTemplate.Style	983
2.243.13	MapMarkerTemplate.ToolTip	983
2.244	MapPointRules	984
2.244.1	MapPointRules.MapColorPaletteRule	984
2.244.2	MapPointRules.MapColorRangeRule	985
2.244.3	MapPointRules.MapCustomColorRule	985
2.244.4	MapPointRules.MapMarkerRule	986
2.244.5	MapPointRules.MapSizeRule	986
2.245	MapMarkerRule	987
2.245.1	MapMarkerRule.BucketCount	988
2.245.2	MapMarkerRule.DataElementName	988
2.245.3	MapMarkerRule.DataElementOutput	989
2.245.4	MapMarkerRule.DataValue	989
2.245.5	MapMarkerRule.DistributionType	990
2.245.6	MapMarkerRule.EndValue	990
2.245.7	MapMarkerRule.LegendName	990
2.245.8	MapMarkerRule.LegendText	991
2.245.9	MapMarkerRule.MapBuckets	991
2.245.10	MapMarkerRule.MapMarkers	992
2.245.11	MapMarkerRule.StartValue	992
2.246	MapMarkers	992
2.246.1	MapMarkers.MapMarker	993
2.247	MapMarker	993
2.247.1	MapMarker.MapMarkerImage	994
2.247.2	MapMarker.MapMarkerStyle	994
2.248	MapMarkerImage	995
2.248.1	MapMarkerImage.MIMETYPE	996
2.248.2	MapMarkerImage.ResizeMode	996
2.248.3	MapMarkerImage.Source	996
2.248.4	MapMarkerImage.TransparentColor	997
2.248.5	MapMarkerImage.Value	997
2.249	MapPoints	998
2.249.1	MapPoints.MapPoint	998
2.250	MapPoint	999
2.250.1	MapPoint.MapFields	999
2.250.2	MapPoint.MapMarkerTemplate	1000
2.250.3	MapPoint.UseCustomPointTemplate	1000
2.250.4	MapPoint.VectorData	1001
2.251	MapPolygonLayer	1001
2.251.1	MapPolygonLayer.Name	1003
2.251.2	MapPolygonLayer.DataElementName	1003
2.251.3	MapPolygonLayer.DataElementOutput	1003
2.251.4	MapPolygonLayer.MapBindingFieldPairs	1004
2.251.5	MapPolygonLayer.MapCenterPointRules	1004
2.251.6	MapPolygonLayer.MapMarkerTemplate	1005
2.251.7	MapPolygonLayer.MapDataRegionName	1005
2.251.8	MapPolygonLayer.MapFieldDefinitions	1005
2.251.9	MapPolygonLayer.MapPolygonRules	1006
2.251.10	MapPolygonLayer.MapPolygons	1006
2.251.11	MapPolygonLayer.MapPolygonTemplate	1007
2.251.12	MapPolygonLayer.MapSpatialDataRegion	1007
2.251.13	MapPolygonLayer.MapShapefile	1008
2.251.14	MapPolygonLayer.MapSpatialDataSet	1008
2.251.15	MapPolygonLayer.MaximumZoom	1009
2.251.16	MapPolygonLayer.MinimumZoom	1009
2.251.17	MapPolygonLayer.Transparency	1009

2.251.18	MapPolygonLayer.VisibilityMode.....	1010
2.252	MapPolygonRules .....	1010
2.252.1	MapPolygonRules.MapColorPaletteRule .....	1011
2.252.2	MapPolygonRules.MapColorRangeRule .....	1011
2.252.3	MapPolygonRules.MapCustomColorRule .....	1012
2.253	MapPolygons .....	1013
2.253.1	MapPolygons.MapPolygon .....	1013
2.254	MapPolygon.....	1013
2.254.1	MapPolygon.MapMarkerTemplate .....	1014
2.254.2	MapPolygon.MapFields.....	1015
2.254.3	MapPolygon.MapPolygonTemplate.....	1015
2.254.4	MapPolygon.UseCustomCenterPointTemplate .....	1015
2.254.5	MapPolygon.UseCustomPolygonTemplate.....	1016
2.254.6	MapPolygon.VectorData .....	1016
2.255	MapPolygonTemplate.....	1017
2.255.1	MapPolygonTemplate.ActionInfo .....	1018
2.255.2	MapPolygonTemplate.CenterPointOffsetX.....	1018
2.255.3	MapPolygonTemplate.CenterPointOffsetY .....	1019
2.255.4	MapPolygonTemplate.DataElementLabel .....	1019
2.255.5	MapPolygonTemplate.DataElementName .....	1019
2.255.6	MapPolygonTemplate.DataElementOutput.....	1020
2.255.7	MapPolygonTemplate.Hidden.....	1020
2.255.8	MapPolygonTemplate.Label .....	1021
2.255.9	MapPolygonTemplate.LabelPlacement.....	1021
2.255.10	MapPolygonTemplate.OffsetX .....	1022
2.255.11	MapPolygonTemplate.OffsetY .....	1022
2.255.12	MapPolygonTemplate.ScaleFactor .....	1023
2.255.13	MapPolygonTemplate.ShowLabel .....	1023
2.255.14	MapPolygonTemplate.Style .....	1023
2.255.15	MapPolygonTemplate.ToolTip .....	1024
2.256	MapTileLayer .....	1024
2.256.1	MapTileLayer.Name.....	1025
2.256.2	MapTileLayer.MapTiles.....	1025
2.256.3	MapTileLayer.MaximumZoom .....	1026
2.256.4	MapTileLayer.MinimumZoom .....	1026
2.256.5	MapTileLayer.TileStyle.....	1027
2.256.6	MapTileLayer.Transparency.....	1027
2.256.7	MapTileLayer.UseSecureConnection .....	1027
2.256.8	MapTileLayer.VisibilityMode.....	1028
2.257	MapTiles .....	1028
2.257.1	MapTiles.MapTile .....	1029
2.258	MapTile.....	1029
2.258.1	MapTile.Name .....	1030
2.258.2	MapTile.TileData .....	1030
2.259	MapLegends .....	1030
2.259.1	MapLegends.MapLegend .....	1031
2.260	MapLegend.....	1031
2.260.1	MapLegend.Name .....	1033
2.260.2	MapLegend.ActionInfo .....	1033
2.260.3	MapLegend.AutoFitTextDisabled .....	1034
2.260.4	MapLegend.BottomMargin.....	1034
2.260.5	MapLegend.DockOutsideViewport .....	1034
2.260.6	MapLegend.EquallySpacedItems .....	1035
2.260.7	MapLegend.Hidden .....	1035
2.260.8	MapLegend.InterlacedRows.....	1035
2.260.9	MapLegend.InterlacedRowsColor .....	1036
2.260.10	MapLegend.Layout.....	1036
2.260.11	MapLegend.LeftMargin .....	1037

2.260.12	MapLegend.MapLegendTitle .....	1037
2.260.13	MapLegend.MapLocation .....	1037
2.260.14	MapLegend.MapSize .....	1038
2.260.15	MapLegend.MinFontSize .....	1038
2.260.16	MapLegend.Position .....	1038
2.260.17	MapLegend.RightMargin .....	1039
2.260.18	MapLegend.Style .....	1039
2.260.19	MapLegend.TextWrapThreshold .....	1040
2.260.20	MapLegend.ToolTip .....	1040
2.260.21	MapLegend.TopMargin .....	1041
2.260.22	MapLegend.ZIndex .....	1041
2.261	MapLegendTitle .....	1041
2.261.1	MapLegendTitle.Caption .....	1042
2.261.2	MapLegendTitle.Style .....	1042
2.261.3	MapLegendTitle.TitleSeparator .....	1043
2.261.4	MapLegendTitle.TitleSeparatorColor .....	1043
2.262	MapTitles .....	1044
2.262.1	MapTitles.MapTitle .....	1044
2.263	MapTitle .....	1045
2.263.1	MapTitle.Name .....	1046
2.263.2	MapTitle.ActionInfo .....	1046
2.263.3	MapTitle.Angle .....	1047
2.263.4	MapTitle.BottomMargin .....	1047
2.263.5	MapTitle.DockOutsideViewport .....	1047
2.263.6	MapTitle.Hidden .....	1048
2.263.7	MapTitle.LeftMargin .....	1048
2.263.8	MapTitle.MapLocation .....	1048
2.263.9	MapTitle.MapSize .....	1049
2.263.10	MapTitle.Position .....	1049
2.263.11	MapTitle.RightMargin .....	1050
2.263.12	MapTitle.Style .....	1050
2.263.13	MapTitle.Text .....	1051
2.263.14	MapTitle.TextShadowOffset .....	1051
2.263.15	MapTitle.ToolTip .....	1051
2.263.16	MapTitle.TopMargin .....	1052
2.263.17	MapTitle.ZIndex .....	1052
2.264	MapViewport .....	1052
2.264.1	MapViewport.BottomMargin .....	1054
2.264.2	MapViewport.ContentMargin .....	1054
2.264.3	MapViewport.GridUnderContent .....	1055
2.264.4	MapViewport.LeftMargin .....	1055
2.264.5	MapViewport.MapCoordinateSystem .....	1056
2.264.6	MapViewport.MapLimits .....	1056
2.264.7	MapViewport.MapLocation .....	1056
2.264.8	MapViewport.MapMeridians .....	1057
2.264.9	MapViewport.MapParallels .....	1057
2.264.10	MapViewport.MapProjection .....	1057
2.264.11	MapViewport.MapSize .....	1058
2.264.12	MapViewport.MapCustomView .....	1058
2.264.13	MapViewport.MapDataBindView .....	1059
2.264.14	MapViewport.MapElementView .....	1059
2.264.15	MapViewport.MaximumZoom .....	1059
2.264.16	MapViewport.MinimumZoom .....	1060
2.264.17	MapViewport.ProjectionCenterX .....	1060
2.264.18	MapViewport.ProjectionCenterY .....	1061
2.264.19	MapViewport.RightMargin .....	1061
2.264.20	MapViewport.SimplificationResolution .....	1061
2.264.21	MapViewport.Style .....	1062

2.264.22	MapViewport.TopMargin	1062
2.264.23	MapViewport.ZIndex	1062
2.265	MapCustomView	1063
2.265.1	MapCustomView.CenterX	1063
2.265.2	MapCustomView.CenterY	1064
2.265.3	MapCustomView.Zoom	1064
2.266	MapDataBoundView	1064
2.266.1	MapDataBoundView.Zoom	1065
2.267	MapElementView	1065
2.267.1	MapElementView.LayerName	1066
2.267.2	MapElementView.MapBindingFieldPairs	1066
2.267.3	MapElementView.Zoom	1067
2.268	MapLimits	1067
2.268.1	MapLimits.MaximumX	1068
2.268.2	MapLimits.MaximumY	1068
2.268.3	MapLimits.MinimumX	1068
2.268.4	MapLimits.MinimumY	1069
2.269	MapMeridians	1069
2.269.1	MapMeridians.Hidden	1070
2.269.2	MapMeridians.Interval	1070
2.269.3	MapMeridians.LabelPosition	1071
2.269.4	MapMeridians.ShowLabels	1071
2.269.5	MapMeridians.Style	1071
2.270	MapParallels	1072
2.270.1	MapParallels.Hidden	1072
2.270.2	MapParallels.Interval	1073
2.270.3	MapParallels.LabelPosition	1073
2.270.4	MapParallels.ShowLabels	1074
2.270.5	MapParallels.Style	1074
2.271	CustomData	1074
2.271.1	CustomData.DataColumnHierarchy	1076
2.271.2	CustomData.DataRowHierarchy	1077
2.271.3	CustomData.DataRows	1077
2.271.4	CustomData.DataSetName	1077
2.271.5	CustomData.Filters	1078
2.271.6	CustomData.SortExpressions	1078
2.271.7	CustomData.DataColumnGroupings	1078
2.271.8	CustomData.DataRowGroupings	1079
2.271.9	CustomData.Relationship	1079
2.272	DataColumnHierarchy	1080
2.272.1	DataColumnHierarchy.DataMembers	1080
2.273	DataMembers	1081
2.273.1	DataMembers.DataMember	1081
2.274	DataMember	1082
2.274.1	DataMember.CustomProperties	1083
2.274.2	DataMember.DataMembers	1083
2.274.3	DataMember.Group	1083
2.274.4	DataMember.SortExpressions	1084
2.274.5	DataMember.Subtotal	1084
2.275	DataRowHierarchy	1084
2.275.1	DataRowHierarchy.DataMembers	1085
2.276	DataRows	1085
2.276.1	DataRows.DataRow	1086
2.277	DataRow	1086
2.277.1	DataRow.DataCell	1087
2.278	DataCell	1087
2.278.1	DataCell.DataValue	1088
2.278.2	DataCell.DataSetName	1089

2.278.3	DataCell.Relationships.....	1089
2.279	DataValue .....	1090
2.279.1	DataValue.Name.....	1090
2.279.2	DataValue.Value.....	1090
2.280	DataColumnGroupings .....	1091
2.280.1	DataColumnGroupings.DataGroupings.....	1091
2.281	DataGroupings.....	1092
2.281.1	DataGroupings.DataGrouping .....	1092
2.282	DataGrouping .....	1092
2.282.1	DataGrouping.CustomProperties.....	1093
2.282.2	DataGrouping.DataGroupings .....	1093
2.282.3	DataGrouping.Grouping .....	1094
2.282.4	DataGrouping.Sorting .....	1094
2.282.5	DataGrouping.Static.....	1095
2.282.6	DataGrouping.Subtotal .....	1095
2.283	DataRowGroupings.....	1095
2.283.1	DataRowGroupings.DataGroupings .....	1096
2.284	CustomReportItem .....	1096
2.284.1	CustomReportItem.Name .....	1100
2.284.2	CustomReportItem.Type.....	1101
2.284.3	CustomReportItem.ActionInfo .....	1101
2.284.4	CustomReportItem.AltReportItem.....	1101
2.284.5	CustomReportItem.Bookmark .....	1102
2.284.6	CustomReportItem.CustomData .....	1102
2.284.7	CustomReportItem.CustomProperties.....	1103
2.284.8	CustomReportItem.DataElementName .....	1103
2.284.9	CustomReportItem.DataElementOutput.....	1103
2.284.10	CustomReportItem.DocumentMapLabel .....	1104
2.284.11	CustomReportItem.Height .....	1104
2.284.12	CustomReportItem.Label .....	1105
2.284.13	CustomReportItem.Left .....	1105
2.284.14	CustomReportItem.RepeatWith .....	1105
2.284.15	CustomReportItem.ReportItems .....	1106
2.284.16	CustomReportItem.Style .....	1106
2.284.17	CustomReportItem.ToolTip .....	1106
2.284.18	CustomReportItem.Top .....	1107
2.284.19	CustomReportItem.Visibility .....	1107
2.284.20	CustomReportItem.Width .....	1108
2.284.21	CustomReportItem.ZIndex.....	1108
2.285	ActionInfo .....	1108
2.285.1	ActionInfo.Actions.....	1110
2.286	Actions .....	1111
2.286.1	Actions.Action .....	1111
2.287	Action.....	1112
2.287.1	Action.BookmarkLink .....	1113
2.287.2	Action.Drillthrough.....	1114
2.287.3	Action.Hyperlink .....	1114
2.287.4	Action.Label .....	1114
2.288	Drillthrough.....	1115
2.288.1	Drillthrough.Parameters .....	1115
2.288.2	Drillthrough.ReportName .....	1116
2.289	Custom.....	1116
2.290	CustomProperties.....	1117
2.290.1	CustomProperties.CustomProperty.....	1118
2.291	CustomProperty .....	1118
2.291.1	CustomProperty.Name .....	1119
2.291.2	CustomProperty.Value.....	1120
2.292	PageBreak.....	1120

2.292.1	PageBreak.BreakLocation	1121
2.292.2	PageBreak.Disabled	1122
2.292.3	PageBreak.ResetPageNumber	1122
2.293	Style	1122
2.293.1	Style.BackgroundColor	1128
2.293.2	Style.BackgroundGradientEndColor	1129
2.293.3	Style.BackgroundGradientType	1129
2.293.4	Style.BackgroundHatchType	1130
2.293.5	Style.BackgroundImage	1133
2.293.6	Style.Border	1133
2.293.7	Style.BorderColor	1133
2.293.8	Style.BorderStyle	1133
2.293.9	Style.BorderWidth	1134
2.293.10	Style.BottomBorder	1134
2.293.11	Style.Calendar	1135
2.293.12	Style.Color	1135
2.293.13	Style.CurrencyLanguage	1136
2.293.14	Style.Direction	1136
2.293.15	Style.FontFamily	1137
2.293.16	Style.FontSize	1138
2.293.17	Style.FontStyle	1138
2.293.18	Style.FontWeight	1138
2.293.19	Style.Format	1139
2.293.20	Style.Language	1140
2.293.21	Style.LeftBorder	1140
2.293.22	Style.LineHeight	1141
2.293.23	Style.NumeralLanguage	1141
2.293.24	Style.NumeralVariant	1142
2.293.25	Style.PaddingBottom	1142
2.293.26	Style.PaddingLeft	1143
2.293.27	Style.PaddingRight	1143
2.293.28	Style.PaddingTop	1144
2.293.29	Style.RightBorder	1144
2.293.30	Style.ShadowColor	1144
2.293.31	Style.ShadowOffset	1145
2.293.32	Style.TextAlign	1145
2.293.33	Style.TextDecoration	1146
2.293.34	Style.TextEffect	1146
2.293.35	Style.TopBorder	1147
2.293.36	Style.UnicodeBiDi	1147
2.293.37	Style.VerticalAlign	1148
2.293.38	Style.WritingMode	1148
2.294	BackgroundImage	1149
2.294.1	BackgroundImage.BackgroundRepeat	1150
2.294.2	BackgroundImage.MIMETYPE	1151
2.294.3	BackgroundImage.Position	1151
2.294.4	BackgroundImage.Source	1152
2.294.5	BackgroundImage.TransparentColor	1153
2.294.6	BackgroundImage.Value	1153
2.294.7	BackgroundImage.EmbeddingMode	1154
2.294.8	BackgroundImage.Transparency	1154
2.295	Border	1155
2.295.1	Border.Color	1156
2.295.2	Border.Style	1156
2.295.3	Border.Width	1157
2.296	BorderColor	1157
2.296.1	BorderColor.Bottom	1158
2.296.2	BorderColor.Default	1158



2.296.3	BorderColor.Left	1158
2.296.4	BorderColor.Right	1159
2.296.5	BorderColor.Top	1159
2.297	BorderStyle	1159
2.297.1	BorderStyle.Bottom	1160
2.297.2	BorderStyle.Default	1161
2.297.3	BorderStyle.Left	1161
2.297.4	BorderStyle.Right	1162
2.297.5	BorderStyle.Top	1163
2.298	BorderWidth	1164
2.298.1	BorderWidth.Bottom	1164
2.298.2	BorderWidth.Default	1165
2.298.3	BorderWidth.Left	1165
2.298.4	BorderWidth.Right	1165
2.298.5	BorderWidth.Top	1166
2.299	Parameters	1166
2.299.1	Parameters.Parameter	1167
2.300	Parameter	1167
2.300.1	Parameter.Name	1167
2.300.2	Parameter.Omit	1168
2.300.3	Parameter.Value	1168
2.301	Visibility	1168
2.301.1	Visibility.Hidden	1170
2.301.2	Visibility.ToggleItem	1170
2.302	Classes	1170
2.302.1	Classes.Class	1171
2.303	Class	1171
2.303.1	Class.ClassName	1172
2.303.2	Class.InstanceName	1172
2.304	CodeModules	1172
2.304.1	CodeModules.CodeModule	1173
2.305	EmbeddedImages	1173
2.305.1	EmbeddedImages.EmbeddedImage	1174
2.306	EmbeddedImage	1174
2.306.1	EmbeddedImage.Name	1175
2.306.2	EmbeddedImage.ImageData	1175
2.306.3	EmbeddedImage.MIMETYPE	1175
2.307	ReportParameters	1176
2.307.1	ReportParameters.ReportParameter	1176
2.308	ReportParameter	1176
2.308.1	ReportParameter.Name	1178
2.308.2	ReportParameter.AllowBlank	1178
2.308.3	ReportParameter.DataType	1178
2.308.4	ReportParameter.DefaultValue	1179
2.308.5	ReportParameter.Hidden	1179
2.308.6	ReportParameter.MultiValue	1180
2.308.7	ReportParameter.Nullable	1180
2.308.8	ReportParameter.Prompt	1181
2.308.9	ReportParameter.UsedInQuery	1181
2.308.10	ReportParameter.ValidValues	1182
2.309	ReportParametersLayout	1182
2.309.1	ReportParametersLayout.GridLayoutDefinition	1182
2.310	GridLayoutDefinition	1183
2.310.1	GridLayoutDefinition.NumberOfColumns	1183
2.310.2	GridLayoutDefinition.NumberOfRows	1184
2.310.3	GridLayoutDefinition.CellDefinitions	1184
2.311	CellDefinitions	1184
2.311.1	CellDefinitions.CellDefinition	1185

2.312	CellDefinition	1185
2.312.1	CellDefinition.ColumnIndex	1186
2.312.2	CellDefinition.RowIndex	1186
2.312.3	CellDefinition.ParameterName	1187
2.313	DefaultValue	1187
2.313.1	DefaultValue.DataSetReference	1188
2.313.2	DefaultValue.Values	1188
2.314	DataSetReference	1188
2.314.1	DataSetReference.DataSetName	1189
2.314.2	DataSetReference.LabelField	1189
2.314.3	DataSetReference.ValueField	1190
2.315	Values	1190
2.315.1	Values.Value	1191
2.316	ValidValues	1191
2.316.1	ValidValues.DataSetReference	1192
2.316.2	ValidValues.ParameterValues	1192
2.317	ParameterValues	1192
2.317.1	ParameterValues.ParameterValue	1193
2.318	ParameterValue	1193
2.318.1	ParameterValue.Label	1194
2.318.2	ParameterValue.Value	1194
2.319	Variables	1195
2.319.1	Variables.Variable	1195
2.320	Variable	1196
2.320.1	Variable.Name	1197
2.320.2	Variable.Value	1197
2.320.3	Variable.Value.DataType	1197
2.320.4	Variable.Writable	1198
2.321	DataSources	1198
2.321.1	DataSources.DataSource	1199
2.322	DataSource	1199
2.322.1	DataSource.Name	1200
2.322.2	DataSource.ConnectionProperties	1200
2.322.3	DataSource.DataSourceReference	1200
2.322.4	DataSource.Transaction	1201
2.323	ConnectionProperties	1201
2.323.1	ConnectionProperties.ConnectString	1202
2.323.2	ConnectionProperties.DataProvider	1202
2.323.3	ConnectionProperties.IntegratedSecurity	1202
2.323.4	ConnectionProperties.Prompt	1202
2.324	DataSets	1203
2.324.1	DataSets.DataSet	1203
2.325	DataSet	1204
2.325.1	DataSet.Name	1207
2.325.2	DataSet.AccentSensitivity	1207
2.325.3	DataSet.CaseSensitivity	1208
2.325.4	DataSet.Collation	1208
2.325.5	DataSet.Fields	1214
2.325.6	DataSet.Filters	1214
2.325.7	DataSet.InterpretSubtotalsAsDetails	1214
2.325.8	DataSet.KanatypeSensitivity	1215
2.325.9	DataSet.Query	1215
2.325.10	DataSet.SharedDataSet	1216
2.325.11	DataSet.WidthSensitivity	1216
2.325.12	DataSet.DefaultRelationships	1217
2.325.13	DataSet.NullsAsBlanks	1217
2.325.14	DataSet.CollationCulture	1217
2.326	SharedDataSet	1218

2.326.1	SharedDataSet.SharedDataSetReference	1218
2.326.2	SharedDataSet.QueryParameters	1219
2.327	Fields	1219
2.327.1	Fields.Field	1219
2.328	Field	1220
2.328.1	Field.Name	1221
2.328.2	Field.DataField	1221
2.328.3	Field.Value	1221
2.328.4	Field.Value.DataType	1222
2.328.5	Field.AggregateIndicatorField	1222
2.329	Filters	1223
2.329.1	Filters.Filter	1223
2.330	Filter	1224
2.330.1	Filter.FilterExpression	1225
2.330.2	Filter.FilterValues	1225
2.330.3	Filter.Operator	1225
2.331	FilterValues	1227
2.331.1	FilterValues.FilterValue	1228
2.331.2	FilterValues.FilterValue.DataType	1229
2.332	Query	1229
2.332.1	Query.CommandText	1230
2.332.2	Query.CommandType	1230
2.332.3	Query.DataSourceName	1231
2.332.4	Query.QueryParameters	1231
2.332.5	Query.Timeout	1231
2.333	QueryParameters	1232
2.333.1	QueryParameters.QueryParameter	1232
2.334	QueryParameter	1233
2.334.1	QueryParameter.Name	1234
2.334.2	QueryParameter.Value	1234
2.334.3	QueryParameter.Value.DataType	1235
2.335	DefaultRelationships	1235
2.335.1	DefaultRelationships.DefaultRelationship	1235
2.336	DefaultRelationship	1236
2.336.1	DefaultRelationship.RelatedDataSet	1236
2.336.2	DefaultRelationship.JoinConditions	1237
2.336.3	DefaultRelationship.NaturalJoin	1237
2.337	JoinConditions	1237
2.337.1	JoinConditions.JoinCondition	1238
2.338	JoinCondition	1238
2.338.1	JoinCondition.ForeignKey	1239
2.338.2	JoinCondition.PrimaryKey	1239
2.338.3	JoinCondition.SortDirection	1240
2.339	Relationships	1240
2.339.1	Relationships.Relationship	1241
2.340	Relationship	1241
2.340.1	Relationship.ParentScope	1242
2.340.2	Relationship.NaturalJoin	1243
2.340.3	Relationship.JoinConditions	1243
2.341	Expressions	1243
2.341.1	Expression Syntax	1243
2.341.2	Custom Code References	1244
2.341.3	Built-in Code Namespace References	1244
2.341.4	Data Types	1244
2.341.5	Expression Error Handling	1244
2.341.6	Global Collections	1245
2.341.6.1	Fields	1245
2.341.6.2	Predefined Field Properties	1246

2.341.6.3	Fields Collection in Reports with Multiple Datasets.....	1247
2.341.6.4	Parameters .....	1247
2.341.6.5	ReportItems .....	1247
2.341.6.6	Globals .....	1248
2.341.6.7	ReadOnlyNameValueCollection.....	1248
2.341.6.8	User .....	1249
2.341.6.9	DataSources.....	1249
2.341.6.10	DataSets.....	1249
2.341.6.11	Variables.....	1250
2.341.6.12	Restrictions on Use of Global Collections .....	1250
2.341.6.13	Scopes .....	1252
2.341.7	Aggregate Functions .....	1252
2.341.7.1	Scope .....	1253
2.341.7.2	Recursive .....	1253
2.341.7.3	Recursive Depth .....	1254
2.341.7.4	Function: Level .....	1254
2.341.7.5	Aggregate Function: Sum.....	1254
2.341.7.6	Aggregate Function: Avg.....	1255
2.341.7.7	Aggregate Function: Max .....	1255
2.341.7.8	Aggregate Function: Min .....	1256
2.341.7.9	Aggregate Function: Count.....	1256
2.341.7.10	Aggregate Function: CountDistinct .....	1256
2.341.7.11	Aggregate Function: CountRows .....	1257
2.341.7.12	Aggregate Function: StDev .....	1257
2.341.7.13	Aggregate Function: StDevP.....	1257
2.341.7.14	Aggregate Function: Var .....	1258
2.341.7.15	Aggregate Function: VarP .....	1258
2.341.7.16	Aggregate Function: Union .....	1258
2.341.7.17	Aggregate Function: First.....	1258
2.341.7.18	Aggregate Function: Last .....	1259
2.341.7.19	Aggregate Function: Previous .....	1259
2.341.7.20	Aggregate Function: RunningValue.....	1260
2.341.7.21	Aggregate Function: RowNumber .....	1260
2.341.7.22	Aggregate Function: Aggregate.....	1261
2.341.7.23	Restrictions on Aggregate Usage.....	1261
2.341.7.24	Restrictions on Aggregate Nesting.....	1262
2.341.7.25	Filtering and Aggregates .....	1262
2.341.7.26	Additional Functions .....	1263
2.341.7.26.1	Dynamic Scoping.....	1263
2.341.7.26.2	Function: InScope .....	1263
2.341.7.26.3	Semantic Query Drillthrough.....	1263
2.341.7.26.4	Function: CreateDrillthroughContext .....	1264
2.341.7.26.5	Function: Lookup.....	1264
2.341.7.26.6	Function: LookupSet.....	1264
2.341.7.26.7	Function: MultiLookup.....	1265
2.341.7.26.8	Function: MinValue .....	1266
2.341.7.26.9	Function: MaxValue .....	1266

**3 Structure Examples ..... 1268**

3.1	Report in RDL schema 2008/01.....	1268
3.2	Report in RDL schema 2010/01.....	1272
3.3	Report in RDL schema 2005/01.....	1278
3.4	Column and Line Chart .....	1281
3.5	Doughnut Chart .....	1299
3.6	CustomReportItem .....	1302
3.7	Tablix 1 .....	1307
3.8	Tablix 2 .....	1309
3.9	Tablix 3 .....	1315

3.10	Data	1328
3.11	Subreport	1329
3.12	TextBox	1329
3.13	GaugePanel	1332
3.14	Line	1338
3.15	List	1339
3.16	Rectangle with Image and Textbox	1343
3.17	Map	1345
3.18	Matrix	1354
3.19	Table	1360
<b>4</b>	<b>Security</b>	<b>1374</b>
<b>5</b>	<b>Appendix A: RDL XML Schemas</b>	<b>1375</b>
5.1	RDL XML Schema for Version 2003/10	1375
5.2	RDL XML Schema for Version 2005/01	1397
5.3	RDL XML Schema for Version 2008/01	1423
5.4	RDL XML Schema for Version 2010/01	1465
5.5	RDL XML Schema for Version 2011/01	1521
5.6	RDL XML Schema for Version 2012/01	1525
5.7	RDL XML Schema for Version 2013/01	1526
5.8	RDL XML Schema for Version 2016/01	1528
<b>6</b>	<b>Appendix B: Error Codes</b>	<b>1584</b>
6.1	rsAggregateInFilterExpression	1584
6.2	rsAggregateInGroupExpression	1584
6.3	rsAggregateInQueryParameterExpression	1584
6.4	rsAggregateInReportParameterExpression	1584
6.5	rsAggregateInReportLanguageExpression	1584
6.6	rsAggregateInCalculatedFieldExpression	1584
6.7	rsAggregateofAggregate	1584
6.8	rsAggregateReportItemInBody	1584
6.9	rsBinaryConstant	1584
6.10	rsChartSeriesPlotTypeIgnored	1585
6.11	rsCompilerErrorInExpression	1585
6.12	rsCompilerErrorInCode	1585
6.13	rsCompilerErrorInClassInstanceDeclaration	1585
6.14	rsUnexpectedCompilerError	1585
6.15	rsConflictingRunningValueScopesInMatrix	1585
6.16	rsConflictingRunningValueScopesInTablix	1585
6.17	rsCountRowsInPageSectionExpression	1586
6.18	rsCountStarNotSupported	1586
6.19	rsCountStarRVNotSupported	1586
6.20	rsCustomAggregateAndFilter	1586
6.21	rsDataRegionInDetailList	1586
6.22	rsDataRegionInPageSection	1586
6.23	rsDataRegionInTableDetailRow	1586
6.24	rsDataRegionWithoutDataSet	1586
6.25	rsDataSourceReferenceNotPublished	1587
6.26	rsDuplicateChartAxisName	1587
6.27	rsSpecifiedNonValueAxisName	1587
6.28	rsValueAxisNameNotFound	1587
6.29	rsInvalidTextEffect	1587
6.30	rsInvalidBackgroundHatchType	1587
6.31	rsInvalidBackgroundImagePosition	1587
6.32	rsPageBreakOnGaugeGroup	1587
6.33	rsDuplicateChartLegendCustomItemCellName	1587
6.34	rsDuplicateChartFormulaParameter	1588
6.35	rsDuplicateClassInstanceName	1588

6.36	rsDuplicateDataSourceName .....	1588
6.37	rsInvalidDataSourceNameLength .....	1588
6.38	rsDuplicateEmbeddedImageName .....	1588
6.39	rsInvalidEmbeddedImageNameNotCLSCompliant .....	1588
6.40	rsInvalidEmbeddedImageNameLength .....	1588
6.41	rsDuplicateFieldName .....	1588
6.42	rsDuplicateReportItemName .....	1588
6.43	rsDuplicateReportParameterName .....	1589
6.44	rsDuplicateCaseInsensitiveReportParameterName .....	1589
6.45	rsDuplicateScopeName .....	1589
6.46	rsExpressionMissingCloseParen .....	1589
6.47	rsFieldInPageSectionExpression .....	1589
6.48	rsFieldInReportParameterExpression .....	1589
6.49	rsFieldInQueryParameterExpression .....	1589
6.50	rsFieldInReportLanguageExpression .....	1589
6.51	rsGlobalNotDefined .....	1589
6.52	rsInvalidAction.....	1589
6.53	rsInvalidAggregateScope.....	1590
6.54	rsInvalidAltReportItem.....	1590
6.55	rsInvalidBooleanConstant.....	1590
6.56	rsInvalidCategoryGrouping .....	1590
6.57	rsInvalidCharacterInExpression.....	1590
6.58	rsInvalidChartGroupings .....	1590
6.59	rsInvalidChartSubType.....	1590
6.60	rsInvalidColumnGrouping .....	1590
6.61	rsInvalidCustomAggregateExpression .....	1591
6.62	rsInvalidCustomAggregateScope .....	1591
6.63	rsInvalidCustomPropertyName.....	1591
6.64	rsInvalidDataElementNameNotCLSCompliant .....	1591
6.65	rsInvalidDataSetName .....	1591
6.66	rsInvalidDataSource .....	1591
6.67	rsInvalidDataSourceReference .....	1591
6.68	rsInvalidValidValuesDataSetReference .....	1591
6.69	rsInvalidDefaultValueDataSetReference .....	1591
6.70	rsInvalidDataSetReferenceField.....	1592
6.71	rsInvalidDefaultValue .....	1592
6.72	rsInvalidDefaultValueValues .....	1592
6.73	rsInvalidDetailsDataGrouping .....	1592
6.74	rsInvalidExpressionScope .....	1592
6.75	rsInvalidExpressionScopeDataSet.....	1592
6.76	rsInvalidSortExpressionScope .....	1592
6.77	rsIneffectiveSortExpressionScope.....	1592
6.78	rsInvalidField.....	1593
6.79	rsInvalidFieldNameNotCLSCompliant .....	1593
6.80	rsInvalidFieldNameLength .....	1593
6.81	rsInvalidGroupExpressionScope .....	1593
6.82	rsInvalidGroupNameNotCLSCompliant .....	1593
6.83	rsInvalidHideDuplicateScope.....	1593
6.84	rsInvalidURLProtocol.....	1593
6.85	rsInvalidIntegerConstant.....	1593
6.86	rsInvalidDateTimeConstant .....	1593
6.87	rsInvalidFloatConstant .....	1594
6.88	rsLabelExpressionOnChartScalarAxisIsIgnored .....	1594
6.89	rsInvalidMatrixSubtotalReportItem .....	1594
6.90	rsInvalidNameNotCLSCompliant.....	1594
6.91	rsInvalidNameLength.....	1594
6.92	rsInvalidNumberOfFilterValues.....	1594
6.93	rsInvalidOmittedExpressionScope.....	1594

6.94	rsInvalidOmittedTargetScope.....	1594
6.95	rsInvalidParameterNameNotCLSCompliant .....	1594
6.96	rsInvalidParameterNameLength .....	1595
6.97	rsInvalidPreviousAggregateInMatrixCell .....	1595
6.98	rsInvalidPreviousAggregateInTablixCell .....	1595
6.99	rsInvalidRepeatWith .....	1595
6.100	rsInvalidReportDefinition.....	1595
6.101	rsInvalidReportParameterDependency .....	1595
6.102	rsInvalidRowGrouping.....	1595
6.103	rsInvalidRunningValueAggregate.....	1595
6.104	rsInvalidScopeInMatrix .....	1595
6.105	rsInvalidScopeInTablix.....	1596
6.106	rsInvalidSeriesGrouping .....	1596
6.107	rsInvalidStaticDataGrouping .....	1596
6.108	rsInvalidReportName.....	1596
6.109	rsInvalidReportNameCharacters .....	1596
6.110	rsInvalidReportUri .....	1597
6.111	rsInvalidTargetScope.....	1597
6.112	rsInvalidTextboxInPageSection .....	1597
6.113	rsInvalidReportItemInPageSection .....	1597
6.114	rsInvalidToggleItem .....	1597
6.115	rsInvalidValidValues .....	1597
6.116	rsInvalidMultiValueParameter .....	1597
6.117	rsMapLayerMissingProperty .....	1597
6.118	rsMapPropertyAlreadyDefined .....	1598
6.119	rsMissingAggregateScope.....	1598
6.120	rsMissingChartDataPoints .....	1598
6.121	rsMissingDataSetName .....	1598
6.122	rsMissingMIMEType .....	1598
6.123	rsMissingParameterDefault .....	1598
6.124	rsMultipleGroupExpressionsOnChartScalarAxis .....	1598
6.125	rsMultipleGroupingsOnChartScalarAxis.....	1598
6.126	rsMultiReportItemsInMatrixSection .....	1599
6.127	rsMultiReportItemsInTablixCell .....	1599
6.128	rsMultiReportItemsInPageSectionExpression.....	1599
6.129	rsMultiReportItemsInCustomReportItem .....	1599
6.130	rsMultiStaticCategoriesOrSeries .....	1599
6.131	rsMultiStaticColumnsOrRows .....	1599
6.132	rsNegativeLeftWidth .....	1599
6.133	rsNegativeTopHeight .....	1599
6.134	rsNonAggregateInMatrixCell .....	1600
6.135	rsNonExistingScope.....	1600
6.136	rsNotAReportDefinition .....	1600
6.137	rsNotACurrentReportDefinition.....	1600
6.138	rsOverlappingReportItems.....	1600
6.139	rsReportItemOutsideContainer.....	1600
6.140	rsPageBreakOnMatrixColumnGroup .....	1600
6.141	rsPageBreakOnChartGroup .....	1600
6.142	rsParameterValueNullOrBlank .....	1601
6.143	rsPreviousAggregateInFilterExpression.....	1601
6.144	rsPreviousAggregateInGroupExpression .....	1601
6.145	rsPreviousAggregateInPageSectionExpression.....	1601
6.146	rsPreviousAggregateInQueryParameterExpression.....	1601
6.147	rsPreviousAggregateInReportParameterExpression .....	1601
6.148	rsPreviousAggregateInReportLanguageExpression .....	1601
6.149	rsPreviousAggregateInSortExpression .....	1601
6.150	rsRepeatWithNotPeerDataRegion.....	1601
6.151	rsReportItemInFilterExpression.....	1601

6.152	rsReportItemInGroupExpression .....	1602
6.153	rsReportItemInQueryParameterExpression .....	1602
6.154	rsReportItemInReportParameterExpression .....	1602
6.155	rsReportItemInSortExpression .....	1602
6.156	rsReportItemInReportLanguageExpression .....	1602
6.157	rsReportItemInVariableExpression.....	1602
6.158	rsReportParameterPropertyTypeMismatch .....	1602
6.159	rsRowNumberInFilterExpression.....	1602
6.160	rsRowNumberInPageSectionExpression.....	1602
6.161	rsRowNumberInQueryParameterExpression .....	1602
6.162	rsRowNumberInReportParameterExpression .....	1603
6.163	rsRowNumberInReportLanguageExpression .....	1603
6.164	rsRowNumberInSortExpression .....	1603
6.165	rsRowNumberInVariableExpression .....	1603
6.166	rsRunningValueInFilterExpression.....	1603
6.167	rsRunningValueInGroupExpression .....	1603
6.168	rsRunningValueInPageSectionExpression.....	1603
6.169	rsRunningValueInQueryParameterExpression.....	1603
6.170	rsRunningValueInReportParameterExpression .....	1603
6.171	rsRunningValueInReportLanguageExpression .....	1603
6.172	rsRunningValueInSortExpression .....	1603
6.173	rsRunningValueInVariableExpression .....	1604
6.174	rsScopeInPageSectionExpression .....	1604
6.175	rsStaticGroupingOnChartScalarAxis .....	1604
6.176	rsToggleInPageSection.....	1604
6.177	rsUnsortedCategoryInAreaChart.....	1604
6.178	rsWrongNumberOfMatrixCells .....	1604
6.179	rsWrongNumberOfMatrixColumns.....	1604
6.180	rsWrongNumberOfMatrixRows .....	1604
6.181	rsWrongNumberOfChartDataPoints.....	1605
6.182	rsWrongNumberOfChartSeries .....	1605
6.183	rsWrongNumberOfChartDataPointsInSeries .....	1605
6.184	rsWrongNumberOfDataValues.....	1605
6.185	rsWrongNumberOfParameters .....	1605
6.186	rsWrongNumberOfTableCells .....	1605
6.187	rsMissingDataGrouping .....	1605
6.188	rsWrongNumberOfDataRows.....	1606
6.189	rwWrongNumberOfDataCellsInDataRow .....	1606
6.190	rsInvalidRecursiveAggregate .....	1606
6.191	rsInvalidAggregateRecursiveFlag.....	1606
6.192	rsPostSortAggregateInGroupFilterExpression .....	1606
6.193	rsPostSortAggregateInSortExpression .....	1606
6.194	rsPostSortAggregateInVariableExpression .....	1606
6.195	rsAggregateInPreviousAggregate .....	1606
6.196	rsRunningValueInPreviousAggregate.....	1606
6.197	rsPreviousInPreviousAggregate .....	1607
6.198	rsRowNumberInPreviousAggregate.....	1607
6.199	rsInScopeOrLevelInPreviousAggregate.....	1607
6.200	rsInvalidScopeInInnerAggregateOfPreviousAggregate.....	1607
6.201	rsInvalidGroupingParent .....	1607
6.202	rsMissingDataGroupings.....	1607
6.203	rsMissingDataCells .....	1607
6.204	rsCRIMultiStaticColumnsOrRows .....	1607
6.205	rsCRISStaticWithSubgroups .....	1607
6.206	rsCRIMultiNonStaticGroups.....	1608
6.207	rsCRISubtotalNotSupported.....	1608
6.208	rsInvalidGrouping.....	1608
6.209	rsCRIInPageSection.....	1608



6.210	rsBookmarkInPageSection	1608
6.211	rsCantMakeTableGroupHeadersFixed	1608
6.212	rsFixedHeadersInInnerDataRegion	1608
6.213	rsInvalidFixedTableColumnHeaderSpacing	1608
6.214	rsUnsupportedProtocol	1609
6.215	rsVariableInPreviousAggregate	1609
6.216	rsAggregateOfVariable	1609
6.217	rsVariableInQueryParameterExpression	1609
6.218	rsVariableInReportParameterExpression	1609
6.219	rsVariableInReportLanguageExpression	1609
6.220	rsVariableInGroupExpression	1609
6.221	rsVariableInCalculatedFieldExpression	1609
6.222	rsDataSetInPageSectionExpression	1609
6.223	rsDataSetInQueryParameterExpression	1610
6.224	rsDataSetInReportParameterExpression	1610
6.225	rsDataSetInReportLanguageExpression	1610
6.226	rsDataSourceInPageSectionExpression	1610
6.227	rsDataSourceInQueryParameterExpression	1610
6.228	rsDataSourceInReportParameterExpression	1610
6.229	rsDataSourceInReportLanguageExpression	1610
6.230	rsInvalidMeDotValueInExpression	1610
6.231	rsWrongNumberOfTablixCornerRows	1610
6.232	rsWrongNumberOfTablixCornerCells	1610
6.233	rsWrongNumberOfTablixColumns	1611
6.234	rsWrongNumberOfTablixCells	1611
6.235	rsWrongNumberOfTablixRows	1611
6.236	rsInvalidTablixCornerCellSpan	1611
6.237	rsInvalidTablixCornerRowSpans	1611
6.238	rsInvalidTablixCornerColumnSpans	1611
6.239	rsInvalidSortNotAllowed	1612
6.240	rsInvalidFixedHeaderOnOppositeHierarchy	1612
6.241	rsInvalidFixedDataColumnPosition	1612
6.242	rsInvalidFixedDataRowPosition	1612
6.243	rsInvalidFixedDataNotContiguous	1612
6.244	rsInvalidFixedDataInHierarchy	1612
6.245	rsInvalidKeepWithGroup	1612
6.246	rsInvalidKeepWithGroupOnDynamicTablixMember	1613
6.247	rsInvalidKeepWithGroupOnColumnTablixMember	1613
6.248	rsInvalidRepeatOnNewPageOnColumnTablixMember	1613
6.249	rsInvalidRepeatOnNewPage	1613
6.250	rsInvalidTablixCellColSpans	1613
6.251	rsInvalidTablixCellColSpan	1613
6.252	rsInvalidTablixCellRowSpan	1613
6.253	rsCellContentsNotOmitted	1614
6.254	rsCellContentsRequired	1614
6.255	rsInvalidTablixCellCellSpan	1614
6.256	rsInconsistentNumberOfCellsInRow	1614
6.257	rsInvalidTablixHeaderSize	1614
6.258	rsInvalidTablixHeaders	1614
6.259	rsInvalidInnerDataSetName	1614
6.260	rsDuplicateVariableName	1615
6.261	rsInvalidVariableReference	1615
6.262	rsInvalidVariableNameNotCLSCompliant	1615
6.263	rsInvalidVariableNameLength	1615
6.264	rsInvalidGroupingVariableNameNotCLSCompliant	1615
6.265	rsInvalidGroupingVariableNameLength	1615
6.266	rsInvalidVariableCount	1615
6.267	rsMissingExpression	1615

6.268	rsInvalidActionsCount	1616
6.269	rsInvalidFixedDataBodyCellSpans	1616
6.270	rsInvalidEmptyImageReference	1616
6.271	rsFieldReference	1616
6.272	rsInvalidBackgroundRepeat	1616
6.273	rsInvalidBackgroundGradientType	1616
6.274	rsInvalidBorderStyle	1616
6.275	rsInvalidCalender	1616
6.276	rsInvalidCalendarForLanguage	1617
6.277	rsInvalidColor	1617
6.278	rsInvalidDirection	1617
6.279	rsInvalidEmbeddedImageProperty	1617
6.280	rsInvalidFontStyle	1617
6.281	rsInvalidFontWeight	1617
6.282	rsInvalidLanguage	1617
6.283	rsInvalidMeasurementUnit	1617
6.284	rsInvalidMIMEType	1617
6.285	rsInvalidNumeralVariant	1617
6.286	rsInvalidNumeralVariantForLanguage	1618
6.287	rsInvalidSize	1618
6.288	rsInvalidTextAlign	1618
6.289	rsInvalidTextDecoration	1618
6.290	rsInvalidUnicodeBiDi	1618
6.291	rsInvalidVerticalAlign	1618
6.292	rsInvalidWritingMode	1618
6.293	rsNegativeSize	1618
6.294	rsOutOfRangeSize	1618
6.295	rsPageNumberInBody	1619
6.296	rsParameterReference	1619
6.297	rsReportItemReference	1619
6.298	rsDataSetReference	1619
6.299	rsDataSourceReference	1619
6.300	rsErrorLoadingCodeModule	1619
6.301	rsInvalidObjectNameNotUnique	1619
6.302	rsInvalidObjectNameNotCLSCompliant	1619
6.303	rsInvalidSourceSeriesName	1619
6.304	rsInvalidDataSourceNameNotCLSCompliant	1620
6.305	rsDuplicateChartLegendItemName	1620
6.306	rsInvalidEnumValue	1620
6.307	rsInvalidListStyle	1620
6.308	rsInvalidMarkupType	1620
6.309	rsMissingAggregateScopeInPageSection	1620
6.310	rsReportItemInScopedAggregate	1620
6.311	rsPageNumberInScopedAggregates	1620
6.312	rsVariableInDataRowSortExpression	1621
6.313	rsAggregateInDataRowSortExpression	1621
6.314	rsVariableInDataRegionOrDataSetFilterExpression	1621
6.315	rsNestedLookups	1621
6.316	rsLookupInFilterExpression	1621
6.317	rsInvalidLookupScope	1621
6.318	rsLookupOfVariable	1621
6.319	rsReportItemInLookupDestinationOrResult	1622
6.320	rsAggregateInLookupDestinationOrResult	1622
6.321	rsPagePropertyInSubsequentReportSection	1622
6.322	rsReportItemReferenceInPageSection	1622
6.323	rsInvalidColumnsInReportSection	1622
6.324	rsRowNumberInLookupDestinationOrResult	1622
6.325	rsPreviousInLookupDestinationOrResult	1622

6.326	rsLevelCallRecursiveHierarchyBothDimensions .....	1623
6.327	rsInvalidColumnsInBody .....	1623
6.328	rsDuplicateGroupingVariableName.....	1623
6.329	rsNestedCustomAggregate .....	1623
6.330	rsInvalidNestedAggregateScope .....	1623
6.331	rsNestedAggregateScopesFromDifferentAxes .....	1623
6.332	rsIncompatibleNestedAggregateScopes.....	1624
6.333	rsNestedAggregateScopeRequired .....	1624
6.334	rsInvalidNestedDataSetAggregate .....	1624
6.335	rsDataSetAggregateOfAggregates .....	1624
6.336	rsInvalidNestedRecursiveAggregate.....	1624
6.337	rsRecursiveAggregateOfAggregate .....	1624
6.338	rsPostSortAggregateInAggregateExpression .....	1624
6.339	rsRunningValueInAggregateExpression .....	1625
6.340	rsPreviousInAggregateExpression.....	1625
6.341	rsNestedAggregateViaLookup .....	1625
6.342	rsNestedAggregateInPageSection .....	1625
6.343	rsNestedAggregateInFilterExpression.....	1625
6.344	rsNestedAggregateInGroupVariable .....	1625
6.345	rsVariableTypeNotSerializable .....	1625
6.346	rsInvalidWritableVariable .....	1626
6.347	rsOverallPageNumberInScopedAggregate.....	1626
6.348	rsOverallPageNumberInBody .....	1626
6.349	rsParameterPropertyTypeMismatch .....	1626
6.350	rsStateIndicatorInvalidTransformationScope.....	1626
6.351	rsVariableInJoinExpression .....	1626
6.352	rsReportItemInJoinExpression .....	1626
6.353	rsRunningValueInJoinExpression .....	1626
6.354	rsPreviousAggregateInJoinExpression .....	1627
6.355	rsAggregateInJoinExpression .....	1627
6.356	rsElementMustContainChildren.....	1627
6.357	rsElementMustContainChild .....	1627
6.358	rsMissingDefaultRelationshipJoinCondition .....	1627
6.359	rsNonExistingRelationshipRelatedScope .....	1627
6.360	rsInvalidSelfJoinRelationship.....	1627
6.361	rsInvalidDefaultRelationshipNotNaturalJoin.....	1627
6.362	rsInvalidRelationshipGroupingContainerNotNaturalGroup .....	1628
6.363	rsInvalidRelationshipContainerNotNaturalJoin .....	1628
6.364	rsInvalidDefaultRelationshipDuplicateRelatedDataset .....	1628
6.365	rsInvalidDefaultRelationshipCircularReference.....	1628
6.366	rsInvalidRelationshipDataSetUsedMoreThanOnce .....	1628
6.367	rsInvalidRelationshipDataSet .....	1628
6.368	rsInvalidNaturalSortContainer.....	1628
6.369	rsInvalidSortingContainerNotNaturalSort.....	1629
6.370	rsConflictingNaturalSortRequirements.....	1629
6.371	rsIncompatibleNaturalSortAndNaturalGroup .....	1629
6.372	rsInvalidNaturalSortFlagCombination .....	1629
6.373	rsInvalidGroupingNaturalGroupFeature .....	1629
6.374	rsInvalidGroupingContainerNotNaturalGroup .....	1629
6.375	rsConflictingNaturalGroupRequirements .....	1629
6.376	rsInvalidBandInvalidLayoutDirection .....	1630
6.377	rsInvalidBandPageBreakIsSet .....	1630
6.378	rsInvalidBandShouldNotBeTogglable.....	1630
6.379	rsInvalidBandNavigationReference .....	1630
6.380	rsInvalidBandNavigationItem .....	1630
6.381	rsInvalidBandNavigations .....	1630
6.382	rsInvalidSliderDataSetReference .....	1631
6.383	rsInvalidSliderDataSetReferenceField.....	1631

6.384	rsBandKeepTogetherIgnored .....	1631
6.385	rsBandIgnoredProperties.....	1631
6.386	rsCollationAndCollationCultureSpecified .....	1631
6.387	rsInvalidAggregateIndicatorField .....	1631
6.388	rsAggregateIndicatorFieldOnCalculatedField .....	1632
6.389	rsInvalidSortDirectionMustNotBeSpecified .....	1632
6.390	rsInvalidNaturalCrossJoin .....	1632
6.391	rsInvalidIntersectionNaturalCrossJoin .....	1632
6.392	rsMissingIntersectionRelationshipParentScope .....	1632
6.393	rsInvalidRelationshipDuplicateParentScope .....	1632
6.394	rsInvalidCellDataSetName .....	1633
6.395	rsDefaultRelationshipIgnored .....	1633
6.396	rsMissingIntersectionDataSetName .....	1633
6.397	rsInvalidRelationshipTopLevelDataRegion .....	1633
6.398	rsConflictingSortFlags .....	1633
6.399	rsInvalidSortFlagCombination .....	1633
6.400	rsInvalidDeferredSortContainer .....	1633
6.401	rsDuplicateReportSectionName .....	1634
6.402	rsInvalidFeatureRdlAttribute .....	1634
6.403	rsSerializableTypeNotSupported .....	1634
6.404	rsInvalidScopeReference .....	1634
6.405	rsInvalidScopeCollectionReference .....	1634
6.406	rsScopeReferenceInComplexExpression .....	1634
6.407	rsScopeReferenceUsesDataSetMoreThanOnce .....	1634
<b>7</b>	<b>Appendix C: Product Behavior .....</b>	<b>1635</b>
<b>8</b>	<b>Change Tracking.....</b>	<b>1649</b>
<b>9</b>	<b>Index.....</b>	<b>1650</b>

# 1 Introduction

The Report Definition Language (RDL) file format is the file format for Microsoft SQL Server RDL, a file type that is used to represent the metadata for defining a **report**.

Sections 1.7 and 2 of this specification are normative. All other sections and examples in this specification are informative.

## 1.1 Glossary

This document uses the following terms:

**absolute path:** A string that identifies the location of a file and that begins with a drive identifier and root directory or network share and ends with the complete file name. Examples are C:\Documents\Work\example.txt and \\netshare\Documents\Work\example.txt.

**action:** An interactivity event in a report, such as a hyperlink, bookmark link, or drillthrough link, that is associated with an item in a report.

**aggregate function:** A function that returns a single value by summarizing data. RDL specifies the following standard aggregate functions: Sum, Avg, Max, Min, Count, CountDistinct, CountRows, StDev, StDevP, Var, and VarP. In addition, RDL specifies the following advanced aggregate functions: First, Last, Previous, RunningValue, RowNumber, and Aggregate.

**anti-aliasing:** The smoothing of the jagged appearance of font characters and lines, which is an artifact of the limited resolution on an output device. The pixels that surround the edges of the character glyph or line are changed to varying shades of color in order to blend the sharp edge into the background.

**background image:** The background that appears behind an entire **report** or a particular **report item**.

**band:** An interactive **tablix** with a navigation mechanism. Multiple navigation mechanisms are available to choose for navigating through the band sheets.

**band sheet:** A page in a band. A band can have more than one sheet.

**bookmark:** An anchor that is used in a report to assist navigation, typically through the use of hyperlinks. A bookmark link in a report sends the user to another location in the report.

**border skin:** The appearance of the border that is drawn around a **chart** or map.

**chart:** An object that displays data or the relationships between sets of data in a visual form.

**chart area:** A region in a chart object that is used to position chart elements, render axes, and plot data.

**chart data region:** A report item on a report layout that displays data in a graphical format.

**chart item:** An item within a **chart** such as an axis, legend, series, data point, data label, trendline, error bar, or data table.

**class:** A reference to a class module whose methods and properties can be used within a report.

**CLS-compliant identifier:** An identifier that must follow the rules of Annex 7 of Technical Report 15 of the Unicode Standard 3.0 [UTR15], which governs the set of characters that can start and be included in identifiers. For two identifiers to be considered distinct, they must differ by more than just their case.

**color palette:** A collection of colors that is available to format text, shapes, cells, and chart elements.

**column header:** The header area of a column in a data region layout.

**column member:** A member in a data region layout that represents a column, including an optional header, an optional grouping of data, and an optional inner hierarchy.

**coverflow:** A mechanism that is used for navigating among band sheets.

**custom property:** Data that can be passed from a **report** for use by a custom report item or a custom user-defined renderer.

**custom report item:** A **report item** that provides a way to add certain types of report functionality that is not natively supported in the RDL schema.

**data correlation:** A **join** between **related datasets**.

**data element:** An element that describes how **report** data is to be represented in a text rendering.

**data point:** A prototype for the data values that are displayed by a chart series. Data points can be displayed in different shapes depending on the **chart** type.

**data provider:** A known data source that is specific to a target type and that provides data to a collector type.

**data region:** A region of a table that encompasses the range of cells that contains the table records. A data region does not include the header row, insert row, or total row of a table.

**data rendering:** A rendering of a **report** that is intended for data processing by a machine and that does not necessarily contain any visual information for human reading.

**data scope:** An RDL data scope: **dataset**, **data region**, dynamic member, and cell.

**data visualization data region:** A **data region** for a **chart** or **gauge panel**.

**dataset:** A named specification that includes a data source definition, a query definition, and optional parameter values, calculated fields, and filtering and collation information as part of a report definition (.rdl) file. An .rdl file can have multiple datasets.

**derived series:** A set of data points whose values are calculated by using statistical or financial formulas. Also called "calculated series".

**detail group:** A Group element that has no descendant group expressions.

**detail member:** A dynamic member that has no descendant group expressions.

**detail scope:** A scope represented by a Group element that has no descendant group expression.

**deviceinfo:** A string key/value pair of configuration information that is passed to a renderer.

**document map:** A UI element that provides a hierarchical listing of report items and group labels within the UI, and that reflects the object containment hierarchy in the report definition. The preferred listing for peer items is left-to-right top-to-bottom order.

**drillthrough parameter:** A parameter that can be passed to a **drillthrough report**.

**drillthrough report:** A report that can be executed and viewed after clicking a report item that contains an action that has a drillthrough link.

**dynamic category:** A dynamic member in the ChartCategoryHierarchy element or DynamicCategories element.

**dynamic column:** A **member** in a **data region** layout that represents a column and repeats based on its data grouping.

**dynamic group:** A Group element that has at least one descendant group expression.

**dynamic member:** A member that has a Group element descendant. See also **detail member** and **static member**.

**dynamic row:** A **member** in a **data region** layout that represents a row and repeats based on its data grouping.

**dynamic series:** A **dynamic member** in the ChartSeriesHierarchy element or DynamicSeries element.

**embedded image:** An **image** that is stored within a document rather than being linked to a source file that is outside the document.

**empty point:** A data point that does not have values for the value axis.

**filter:** A set of criteria that controls the set of records that is returned as a result set.

**gauge:** A presentation of data as a range with an indicator that points to a specific value within the range.

**gauge panel:** The encompassing outer element that contains one or more radial gauges and linear gauges. The gauge panel is used to define properties that apply to all gauges in a gauge panel.

**global collection:** A collection of predefined property names accessible to all expressions in a report.

**group expression:** An expression by which to group the data.

**grouping scope:** A **column member** or **row member** in a **data region** layout that has a non-empty descendant group element.

**image:** A rectangular grid of pixels that has a definite height and a definite width and that can be displayed in a report.

**interactive renderer:** A renderer for reports that can also output interactivity features such as actions.

**intersection scope:** The scope at the intersection of a dynamic column group and a dynamic row group, such as TablixCell, DataCell, and ChartDataPoint.

**join:** A combination of data from multiple sources that is based upon a common attribute or join condition.

**leaf level:** The bottom level of a **data region** member hierarchy. A leaf **member** does not have children.

**line:** A standalone segment that connects two points within a **report**.

**line layer:** A map layer that displays line-based spatial elements.

**line template:** An appearance template that is applied to map lines in a mapline layer.

**linear gauge:** A gauge that is rectangular in shape and can contain one or more linear pointers.

**linear pointer:** A vertically or horizontally facing pointer that points to a value in a linear gauge.

**linear scale:** A vertically or horizontally aligned scale that can cover a range of values within a linear gauge.

**list:** A **report item** on a report layout that displays data in a list format.

**list data region:** A **report item** on a report layout that displays data in a list format.

**map:** A **report item** on a report layout that displays data as a graphical map visualization.

**map bucket:** A **report item** on a report layout that displays data as a graphical map visualization.

**map color scale:** A scale that displays the range of colors that are used for data distribution in a map.

**map coordinate:** The coordinate system that is used for map contents.

**map distance scale:** A scale that displays the distance scale in a map.

**map item:** A spatial item (map polygon, map point, or map line) that is displayed in a map layer.

**map layer:** A layer that contains map items in a map.

**map line:** A visual element in a map that displays a line-based spatial element.

**map marker:** A visual element that is used to mark a map point within a map.

**map meridian:** Vertical gridlines that are displayed in a map.

**map parallel:** The horizontal gridlines that are displayed in a map.

**map point:** A visual element in a map that displays a point-based spatial element.

**map polygon:** A visual element in a map that displays a polygon-based spatial element.

**map tile:** An **image** that displays a portion of a tile layer.

**map viewport:** A rectangular area that shows the content of a map.

**marker template:** A template that is applied to map markers in a polygon layer or a point layer.

**matrix:** A **report item** on a report layout that displays data in a variable columnar format.

**matrix data region:** A report item on a report layout that displays data in a variable columnar format.

**member:** A general term for a **tablix** member, data member, chart member, or gauge member.

**MIME type:** A method that is used by protocol clients to associate files of a certain type with applications that can open or access files of that type.

**null value:** A term that means "having no explicitly assigned value." In particular, a null value is different from a zero or a blank.

**page:** Represents the layout settings for page-oriented report rendering formats.

**page footer:** A set of static text, **images**, lines, rectangles, borders, background color, and background images that repeats on the bottom of each page of a **report**.

**page header:** A set of static text, **images**, lines, rectangles, borders, background color, and background images that repeats on the top of each page of a **report**.



**physical-page oriented renderer:** A renderer that generates page breaks based on physical measurements. The renderer can set properties to precisely control how the pages will break in **reports**.

**play axis:** A mechanism for navigating among band sheets.

**plot area:** A portion of a chart area that contains the plotted data and axes.

**point layer:** A map layer that displays point-based spatial elements.

**pointer image:** An **image** that is used as a pointer within a gauge.

**polygon layer:** A map layer that displays polygon-based spatial elements.

**polygon template:** An appearance template that is to be applied to map polygons within a polygon layer.

**query:** A dataset query instance.

**radial gauge:** A gauge that is circular in shape and that can contain one or more radial pointers.

**radial pointer:** An outward pointer that points to a value within a radial gauge.

**radial scale:** A circular or arched scale that can cover a range of values within a radial gauge.

**rectangle:** A **report item** that can be used as a container for multiple report items or as a graphical element on a **report**.

**related dataset:** A **dataset** that with another dataset defines a relationship.

**relationship:** A combination of **related dataset** and **join** conditions to perform **data correlation**.

**relative path:** A string that specifies a folder path or a web directory (such as "/details") relative to either the current report directory (such as "/salesreports") or the current URL for the report server (such as "http://reportserver").

**renderer:** An application that outputs representations of **reports**, which contain visual layout information and/or data to be used by consumers of the **reports**.

**report:** An object that is a combination of three kinds of information: data or other kinds of information about how to obtain the data (queries) as well as the structure of the data; layout or formatting information that describes how the data is presented; and properties of the report, such as author of the report, report parameters, and images included in the report.

**report definition:** The blueprint for a report before the report is processed or rendered. A report definition contains information about the query and layout for the report.

**report item:** An object that exists on a report layout.

**report layout:** A region in a report that consists of a body area and optional page header and page footer areas.

**report parameter:** A parameter that is defined within a **report definition**. All query parameters have report parameters, but report parameters can be created independent of a **query**.

**report parameter layout grid:** A grid that is used to define a layout for **report parameters**. All report parameters are laid out on this grid regardless of their visibility when the report is rendered.

**report section:** A section within a report layout.

**row header:** The header area of a row in a **data region** layout.

**row member:** A member in a data region layout that represents a row, including an optional header, an optional grouping of data, and an optional inner hierarchy.

**rowset:** A set of rows in which each row has columns of data.

**scale break:** An intentional break on the value axis that is meant to improve the readability of data points on a **chart** when there are large differences between data point values.

**scope:** An item that represents a hierarchy in a **report**. There are explicit scopes (such as **data region**, **dataset**, group) and implicit scopes (such as report scope). At any level in the hierarchy, there can be only one ancestor scope (except for the top-level report scope and the **page** scope) but an unlimited number of descendants as well as peer scopes.

**server subtotal:** An aggregate that is already precalculated in the retrieved **dataset**.

**shapefile:** A geospatial vector file format that is used for storing geometric information.

**slider:** A navigation control for a band.

**slider metadata:** A **dataset** that contains identifier (Key) and Label data for every single band sheet.

**spatial element:** An element that is displayed in a map layer.

**static category:** A static member in the ChartCategoryHierarchy element or StaticCategories element.

**static column:** A **member** in a **data region** layout that represents a column that does not have a Group element descendant.

**static member:** A **member**, such as a TablixMember or ChartMember, that does not have a Group element descendant. See also dynamic member.

**static row:** A **member** in a **data region** layout that represents a row that does not have a Group element descendant.

**static series:** A static member in the ChartSeriesHierarchy element or StaticSeries element.

**StringArray:** An array of string.

**strip line:** Vertical and horizontal lines that are used to highlight specific sections of a chart area.

**table:** A data region on a report layout that displays data in a columnar format.

**table group:** A group within a **table**.

**tablix:** A data region that contains rows and columns that resembles a table or matrix, possibly sharing characteristics of both.

**tabstrip:** A mechanism to navigate among band sheets.

**text box:** A **report item** that represents textual information with formatting.

**tile layer:** A layer that displays an image-based map in a map viewport.

**toggle image:** The initial hidden or displayed state, represented by an **image** that displays +/-, for a **report item** that has a toggle item.

**variant:** A value that is specified as one of the following RDL types: String, Integer, DateTime, Float, or Binary.

**VariantArray:** An array of objects.

**Well-Known Binary:** A data format that provides a portable representation of a geometric values as a contiguous stream of bytes as defined by [\[ISO19125-2\]](#).

**zIndex:** Specifies the drawing order of an item within the containing item. Items are drawn in order according to the zIndex element for the item. Items with lower indices are drawn first, appearing behind items with higher indices.

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as defined in [\[RFC2119\]](#). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## 1.2 References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](#).

### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information.

[ECMA-335] ECMA, "Common Language Infrastructure (CLI): Partitions I through VI", Standard ECMA-335, <http://www.ecma-international.org/publications/standards/Ecma-335.htm>

[ECMA-376-2/2] ECMA, "Information technology – Document description and processing languages – Office Open XML File Formats – Part 2: Open Packaging Conventions", 2nd edition, Standard ECMA-376-2, December 2008, [https://www.ecma-international.org/wp-content/uploads/ECMA-376\\_2nd\\_edition\\_december\\_2008.zip](https://www.ecma-international.org/wp-content/uploads/ECMA-376_2nd_edition_december_2008.zip)

[IEEE754] IEEE, "IEEE Standard for Binary Floating-Point Arithmetic", IEEE 754-1985, October 1985, <http://ieeexplore.ieee.org/servlet/opac?punumber=2355>

[ISO19125-2] ISO, "Geographic information—Simple feature access—Part 2: SQL option", ISO 19125-2:2004, [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=40115](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40115)

**Note** There is a charge to download the specification.

[ISO3166-1] ISO, "ISO 3166-1 decoding table", ISO 3166-1, [http://www.iso.org/iso/iso-3166-1\\_decoding\\_table](http://www.iso.org/iso/iso-3166-1_decoding_table)

[ISO639-2] ISO, "Codes for the representation of names of languages -- Part 2: Alpha-3 code", ISO 639-2:1998, [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=4767](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=4767)

**Note** There is a charge to download the specification.

[ISO8601] ISO, "Data elements and interchange formats - Information interchange - Representation of dates and times", ISO 8601:2004, December 2004, [http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_detail.htm?csnumber=40874](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40874)

**Note** There is a charge to download the specification.

[MS-RDLRS] Microsoft Corporation, "[Report Definition Language Report State File Format](#)".

[MS-RSWSRE2005] Microsoft Corporation, "[Report Server Web Service for Report Execution: ReportExecution2005](#)".

[MS-RSWSRMNM2005] Microsoft Corporation, "[Report Server Web Service for Report Management for Native Mode: ReportService2005](#)".

[MS-RSWSRMSM2006] Microsoft Corporation, "[Report Server Web Service for Report Management for SharePoint Mode: ReportService2006](#)".

[MSFT-VBNET] Microsoft Corporation, "Microsoft Visual Basic .NET Language Reference", Redmond: Microsoft Press, 2002, ISBN: 9780735615526.

[OGCSPEC] Open GIS Consortium, "OpenGIS Simple Features Specification For SQL, Revision 1.1", OpenGIS Project Document 99-049, May 1999, [http://portal.opengeospatial.org/files/?artifact\\_id=829](http://portal.opengeospatial.org/files/?artifact_id=829)

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119.html>

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, October 2006, <http://www.rfc-editor.org/rfc/rfc4648.txt>

[UTR15] UNICODE, "UNICODE NORMALIZATION FORMS", Unicode Technical Report #15, November 1999, <https://www.unicode.org/reports/tr15/tr15-18.html>

[XML10/5] Bray, T., Paoli, J., Sperberg-McQueen, C.M., et al., Eds., "Extensible Markup Language (XML) 1.0 (Fifth Edition)", W3C Recommendation, November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126/>

[XMLSCHEMA1/2] Thompson, H., Beech, D., Maloney, M., and Mendelsohn, N., Eds., "XML Schema Part 1: Structures Second Edition", W3C Recommendation, October 2004, <https://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>

[XMLSCHEMA2/2] Biron, P., and Malhotra, A., Eds., "XML Schema Part 2: Datatypes Second Edition", W3C Recommendation, October 2004, <https://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>

[XMLSCHEMA2] Biron, P.V., Ed. and Malhotra, A., Ed., "XML Schema Part 2: Datatypes", W3C Recommendation, May 2001, <https://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>

## 1.2.2 Informative References

[DUNFORM] Dundas Data Visualization, Inc., "Dundas Chart for Windows Forms: Formulas Overview", 2009, <https://origin2.cdn.componentsource.com/sites/default/files/resources/dundas/538236/WinChart2005/FormulasOverview.html>

[MS-DPRDL] Microsoft Corporation, "[Report Definition Language Data Portability Overview](#)".

[MSDN-RDLDT] Microsoft Corporation, "Report Definition Language Data Types", [https://learn.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/bb630466\(v=sql.105\)](https://learn.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/bb630466(v=sql.105))

[MSDN-UTDRA] Microsoft Corporation, "Understanding Tablix Data Region Areas", [https://learn.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/cc645966\(v=sql.105\)](https://learn.microsoft.com/en-us/previous-versions/sql/sql-server-2008-r2/cc645966(v=sql.105))

## 1.3 Overview

This document specifies the file format for RDL, a file type that is used to represent the metadata for defining a **report**. A **report definition** file is an XML file, as specified in [\[XML10/5\]](#).

### 1.3.1 Report Definition Language File Content

SQL Server Report Definition Language (RDL) consists primarily of the following kinds of information:

- **Report data:** Instructions about how to obtain the data to display in the **report** (such as a connection string and **query**), along with information about the structure of that data (such as field names).
- **Calculations:** Calculations and transformations to be performed on the report data, including grouping, sorting, filtering, aggregations, and scalar formulas.
- **Report layout:** Instructions about how to structure, format, and present the data and calculations in the resulting report. This also includes information about the kinds of end-user interactivity to make available in the report.

### 1.3.2 Report Rendering

RDL is output format-neutral. This means that reports that are defined by using RDL can be output or rendered to a variety of formats including web-ready and print-ready formats or data-focused formats such as XML.

RDL itself is not narrowly prescriptive with respect to the output that results from rendering a report. When generating different output formats, products can represent RDL constructs slightly differently or ignore certain constructs completely. For example, a product that generates a textual format can choose to ignore **images** in the report.

### 1.3.3 Document Structure

The root element of an RDL document is the [Report](#) element.

Subelements are order-independent. This includes items in collection elements, unless the collection is explicitly defined as an ordered **list**.

Unless otherwise specified, a subelement can occur once at most as a child of its parent element. A collection can have multiple instances of a subelement. This will be explicitly stated.

White space is not trimmed from values in the RDL document.

#### 1.3.3.1 XML Namespace

The namespace URI for RDL is as follows:

**<http://schemas.microsoft.com/sqlserver/reporting/yyyy/mm/reportdefinition>**

The date component (yyyy/mm) indicates the date of the release of that version of RDL. The date component is used throughout this document to identify one of the following specific versions of RDL:

- [RDL XML Schema for Version 2003/10](#) (RDL200310)
- [RDL XML Schema for Version 2005/01](#) (RDL200501)
- [RDL XML Schema for Version 2008/01](#) (RDL200801)
- [RDL XML Schema for Version 2010/01](#) (RDL201001)
- [RDL XML Schema for Version 2011/01](#) (RDL201101)
- [RDL XML Schema for Version 2012/01](#) (RDL201201)
- [RDL XML Schema for Version 2013/01](#) (RDL201301)
- [RDL XML Schema for Version 2016/01](#) (RDL201601)

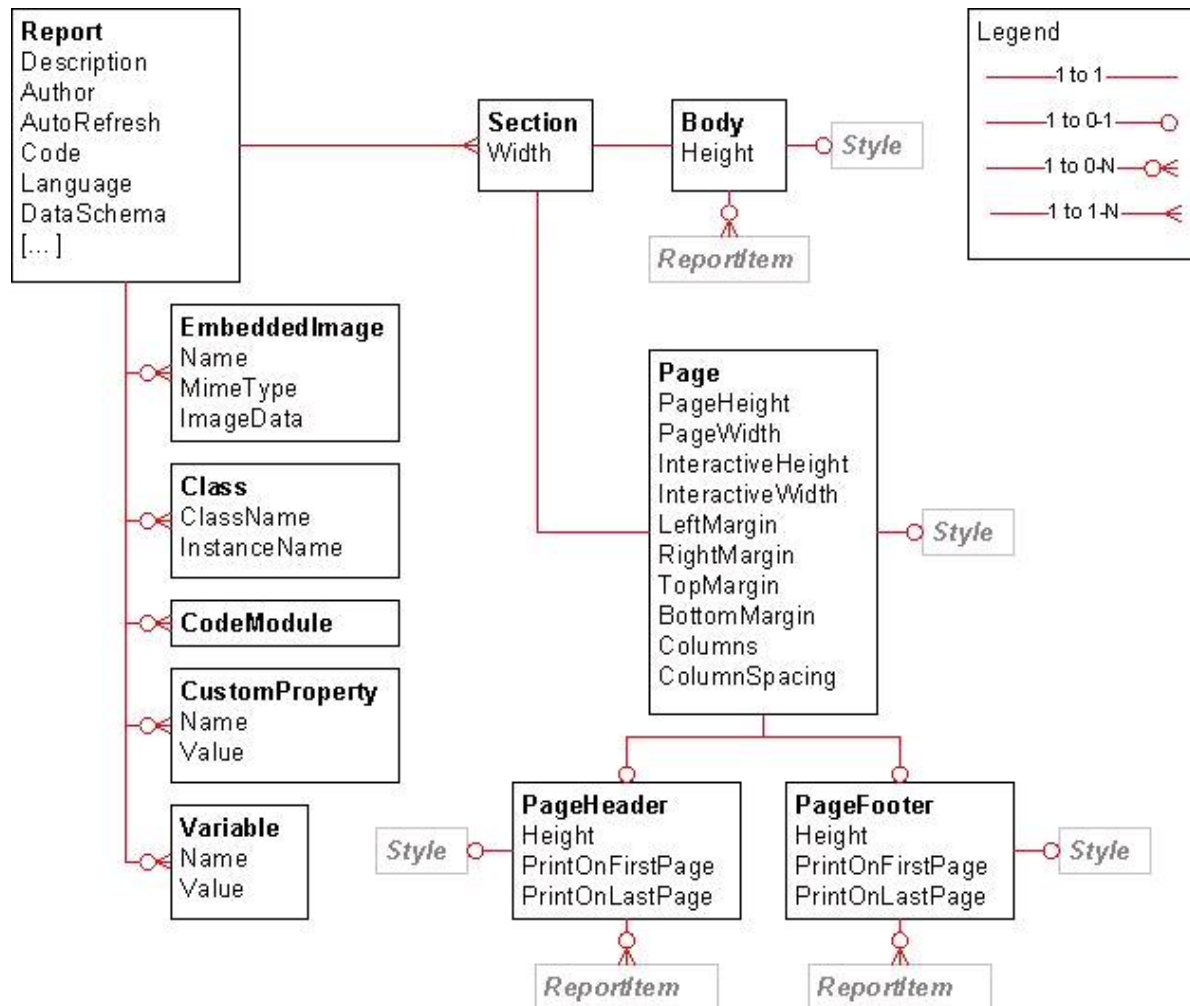
The standard file name extension for RDL files are .rdl and .rdlx.

The **MIME type** to use for RDL files is text/xml.

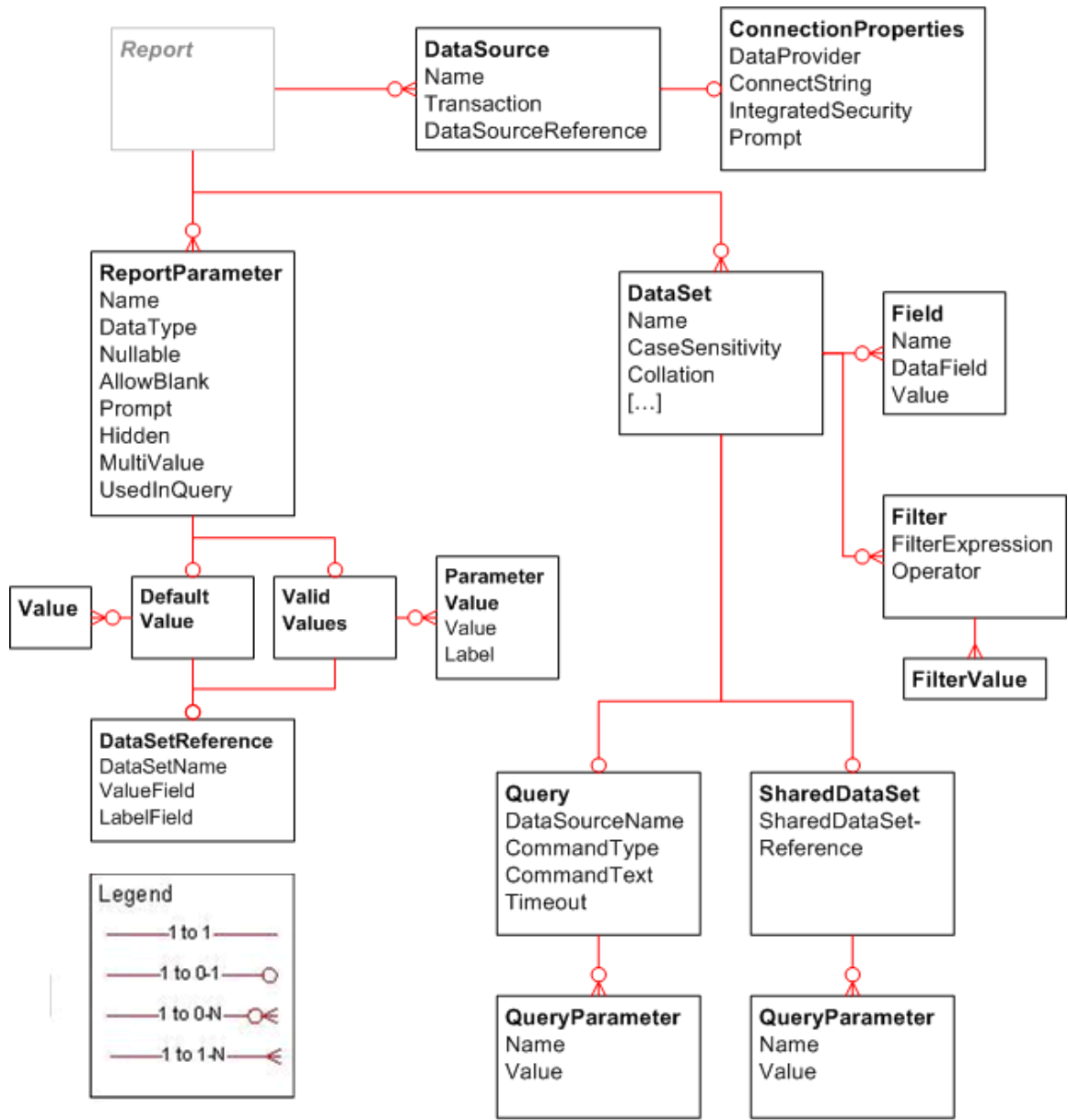
### 1.3.4 Report Definition Overview Diagrams

This section contains diagrams that illustrate the schema of the 2010/01 version of Report Definition Language (section 5.4).

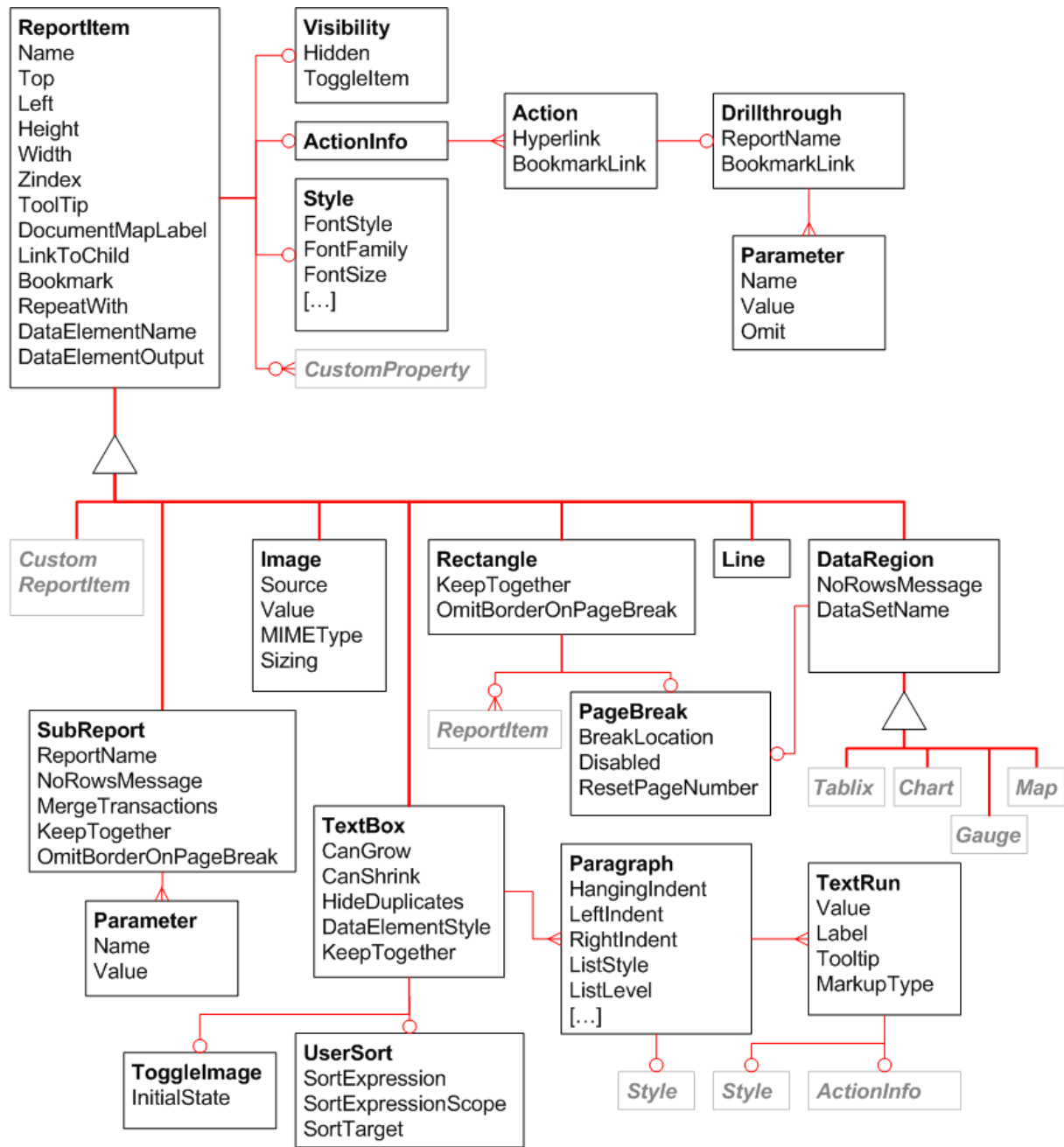
Note that for simplicity, certain related types are represented in the diagrams as abstract base types. For example, the abstract base type **ReportItem** does not appear in RDL. Only concrete derived types (such as **Textbox**) appear in RDL. The following are the abstract base types that are shown in the diagrams: **ReportItem**, **DataRegion**, **Gauge**, **GaugeScale**, and **GaugePointer**.



**Figure 1: Report layout**

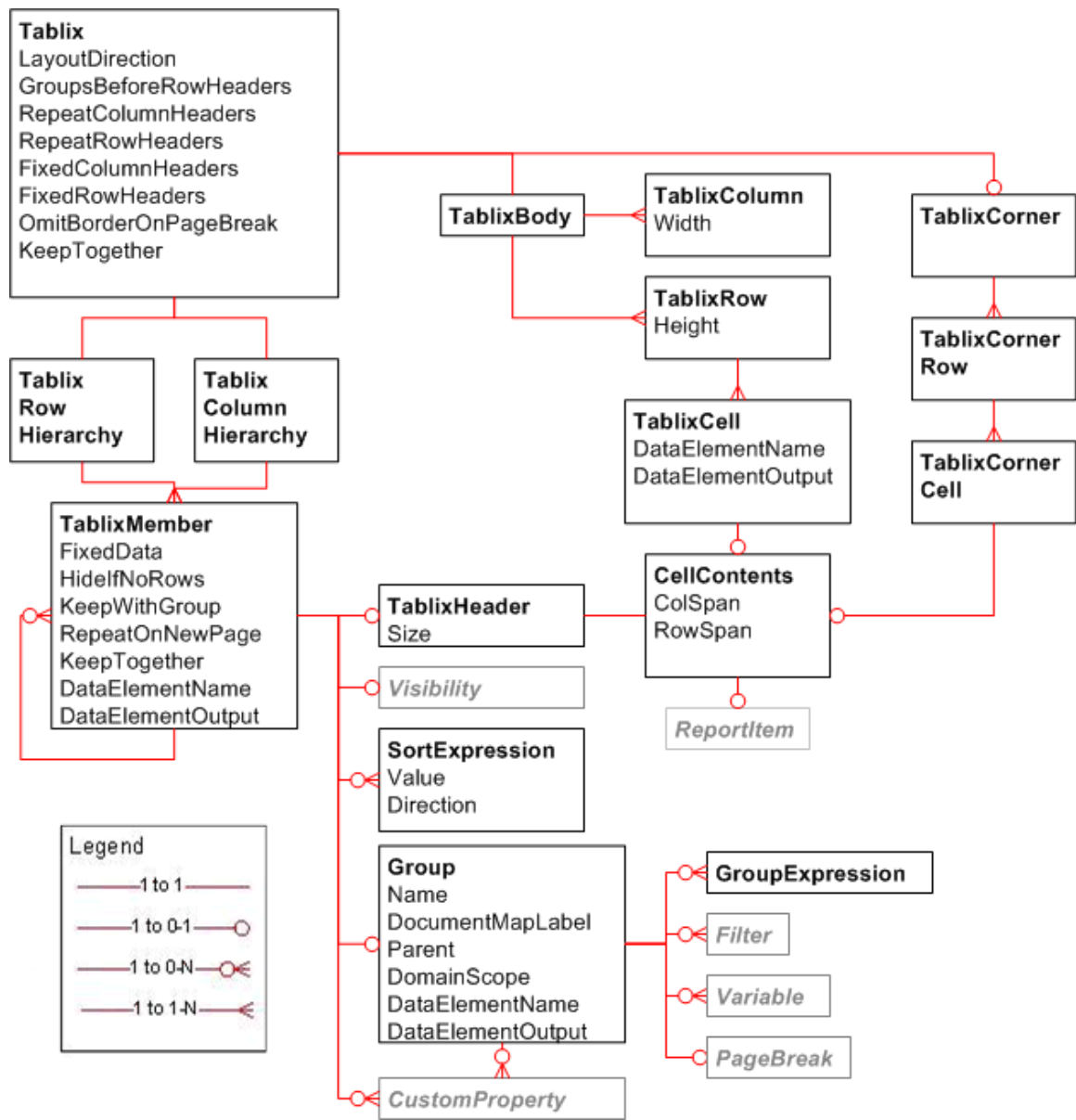


**Figure 2: Report data**

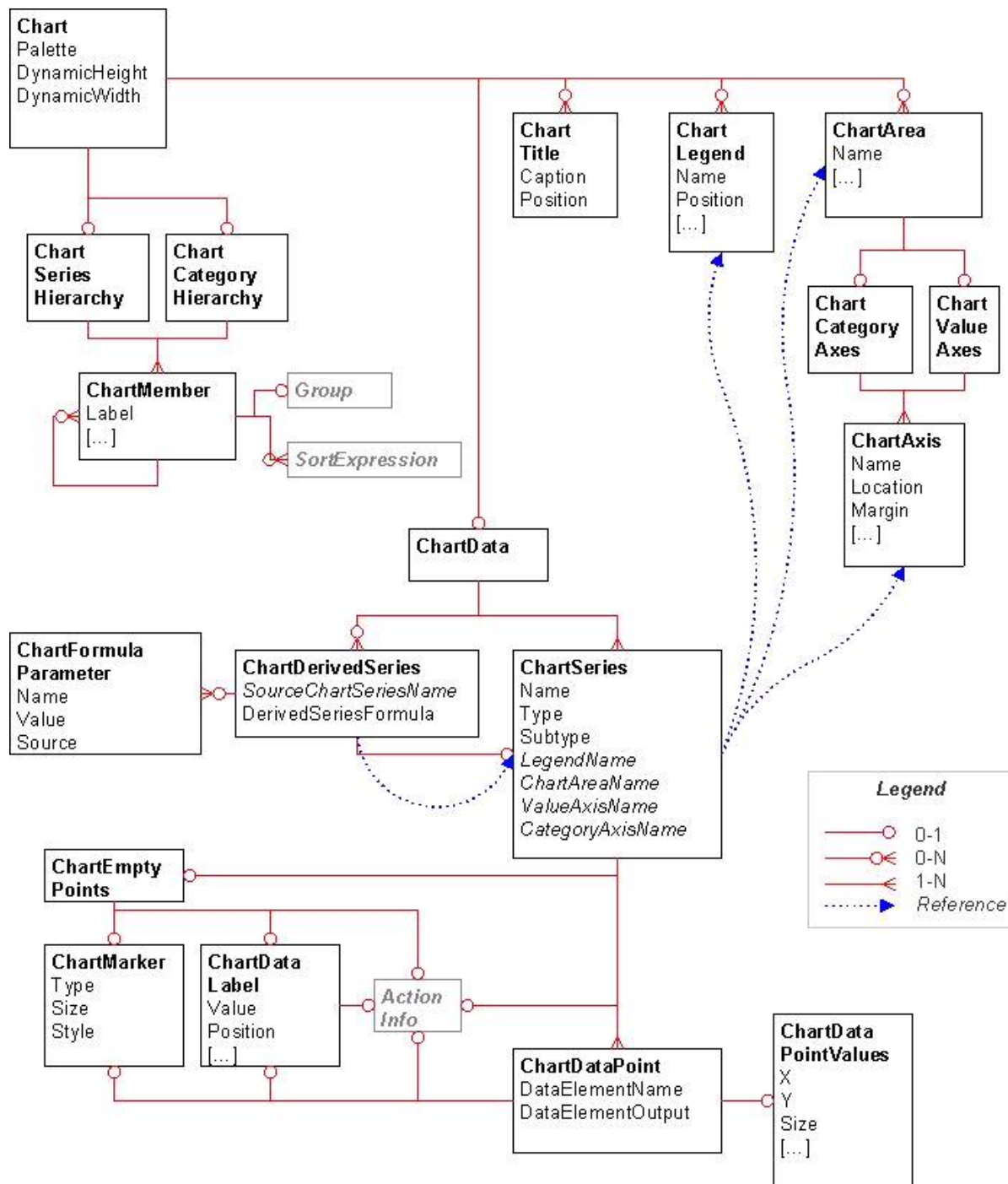


**Figure 3: Report items**

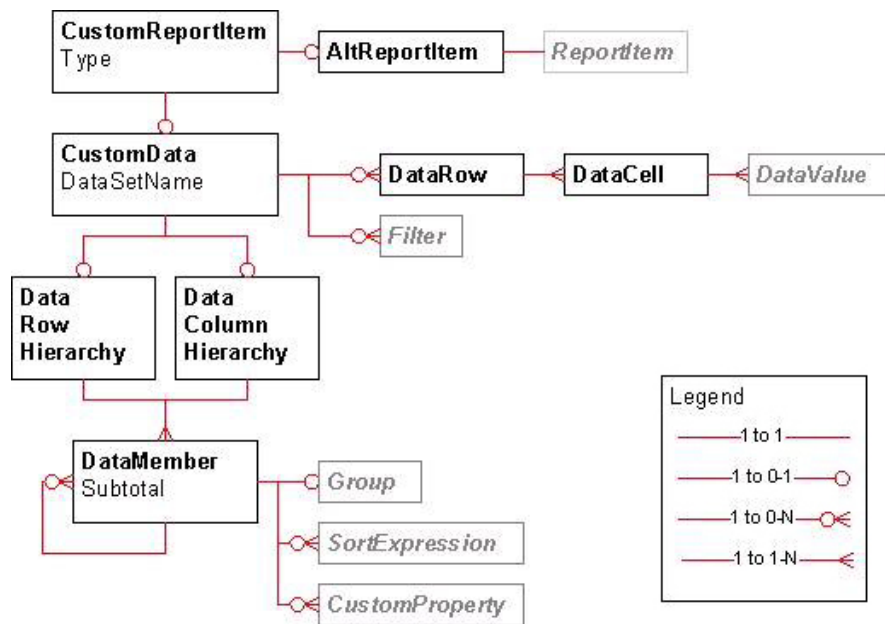




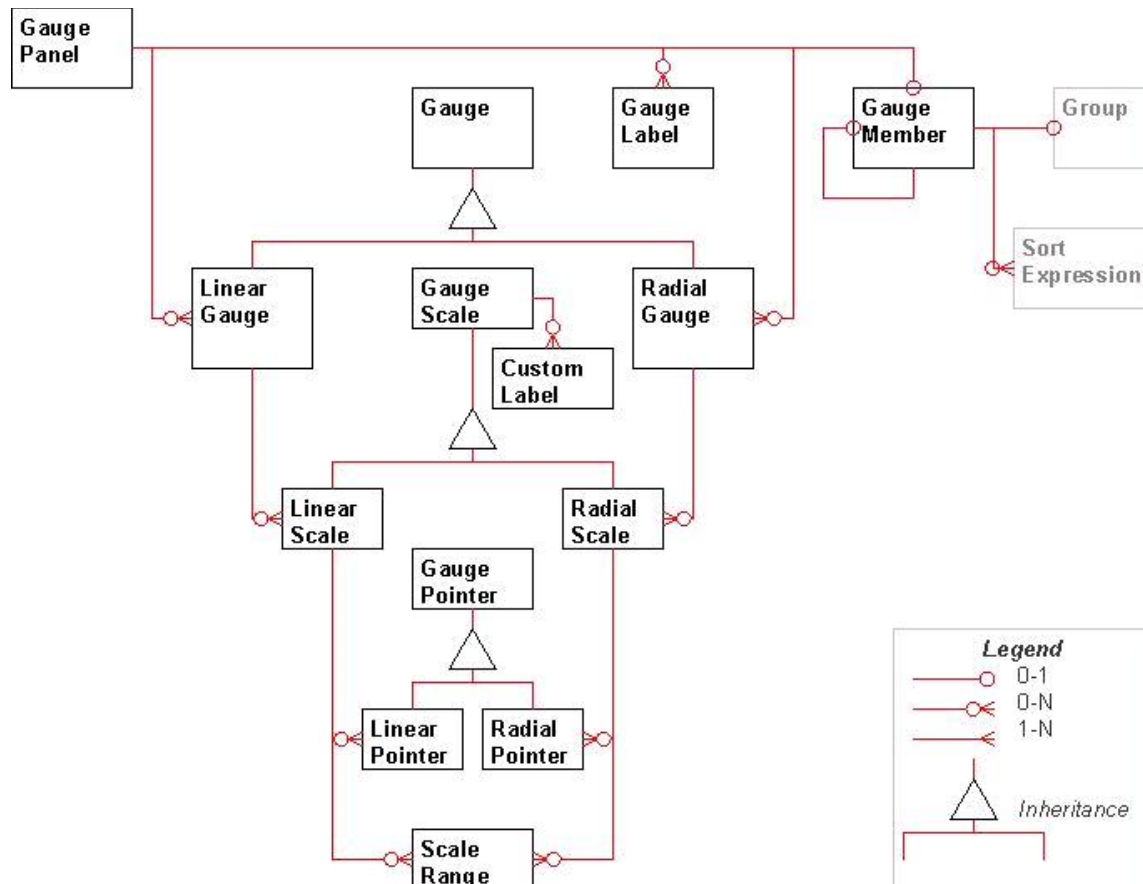
**Figure 4: Tablix**



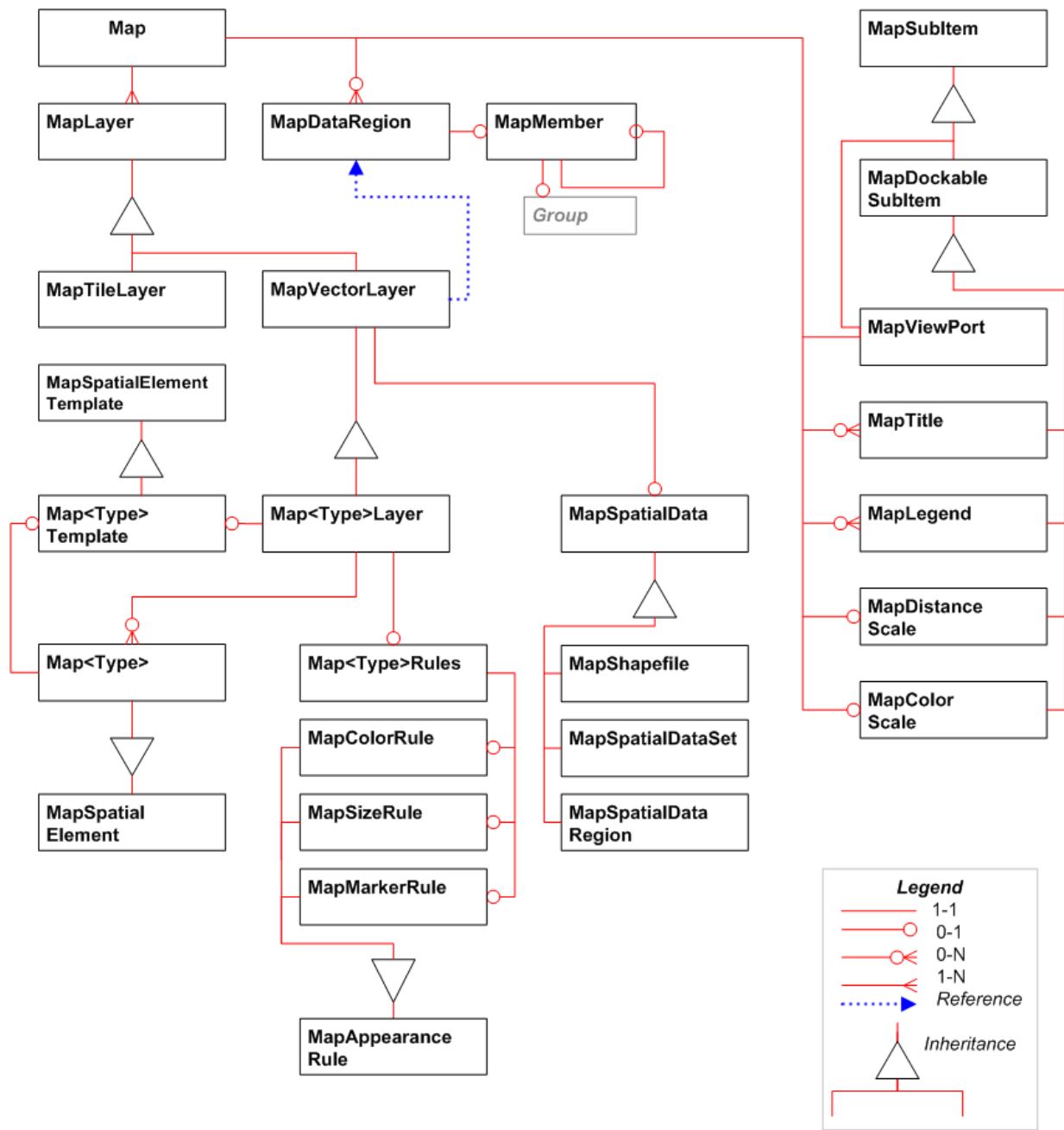
**Figure 5: Chart**



**Figure 6: Custom report item**



**Figure 7: Gauge panel**



<Type> is either Polygon, Line or Point

**Figure 8: Map structures**

### 1.4 Relationship to Protocols and Other Structures

RDL is used as a payload in the following protocols:

- ReportExecution2005 [\[MS-RSWSRE2005\]](#): Used when setting the definition of a report for execution.

- ReportService2005 [\[MS-RSWSRMNM2005\]](#): Used when setting or retrieving the definition of a report in the report catalog.
- ReportService2006 [\[MS-RSWSRMSM2006\]](#): Used when setting or retrieving the definition of a report in the report catalog.

RDL references structures that are defined in [\[MS-RDLRS\]](#).

## 1.5 Applicability Statement

This persistence format is applicable for use as a standalone representation of report metadata and for publishing report metadata from a client to a server.

## 1.6 Versioning and Localization

This document specifies the structures for the following versions of RDL: [<1>](#)

- [RDL XML Schema for Version 2003/10](#) (RDL200310)
- [RDL XML Schema for Version 2005/01](#) (RDL200501)
- [RDL XML Schema for Version 2008/01](#) (RDL200801)
- [RDL XML Schema for Version 2010/01](#) (RDL201001)
- [RDL XML Schema for Version 2011/01](#) (RDL201101)
- [RDL XML Schema for Version 2012/01](#) (RDL201201)
- [RDL XML Schema for Version 2013/01](#) (RDL201301)
- [RDL XML Schema for Version 2016/01](#) (RDL201601)

These versions are defined in [Appendix A: RDL XML Schemas \(section 5\)](#).

There are no localization-dependent structures in the RDL file format.

## 1.7 Vendor-Extensible Fields

RDL is an open schema. Application authors can extend and annotate RDL with their own attributes and subelements (in their own namespace). Note, however, that tools that use RDL can validate the extended namespaces, but do not have to preserve unrecognized elements when loading and persisting. For properties that need to be preserved independent of the tool, the [CustomProperties](#) element can be used instead.

## 2 Structures

### 2.1 Introduction

This section describes the elements that are defined in the Report Definition Language (RDL), including the following:

- The meaning, restrictions, and constraints of the element and its value, if applicable.
- The relationships between the element and other elements.
- The XML schema definition (XSD) of the element.

The RDL XML file is a valid XML file, as specified in [\[XML10/5\]](#), which conforms to one of the XSD specifications that are defined in section [1.3.3.1](#) of this document, depending on the RDL version. Section [5](#) of this document describes the structures of an RDL XML file that conforms to this specification.

The following RDL XML schema versions use a macro-versioning approach:

- RDL 2003/10 (section [5.1](#))
- RDL 2005/01 (section [5.2](#))
- RDL 2008/01 (section [5.3](#))
- RDL 2010/01 (section [5.4](#))
- RDL 2016/01 (section [5.8](#))

Macro-versioning means that each of these RDL schema versions is a complete XML schema that defines all its elements and attributes and the relationships among them. Because each schema is complete in and of itself, each schema supersedes the previous schema, cumulatively.

However, the following RDL schema versions use a micro-versioning approach:

- RDL 2011/01 (section [5.5](#))
- RDL 2012/01 (section [5.6](#))
- RDL 2013/01 (section [5.7](#))

Micro-versioning means that these three RDL schema versions supplement the RDL 2010/01 base schema. Each supplementary schema defines only the incremental set of elements and attributes and the relationships among them that is defined in that particular schema.

Because of the way in which the macro-versioned RDL XML schemas are "open" and allow custom elements and attributes by using the **xsd:any** element, it is not possible to also encode the relationship between RDL 2010/01 elements and RDL 2011/01, RDL 2012/01, and RDL 2013/01 elements across the different schema approaches. Instead, the relationships are validated in the particular code.

As such, the RDL 2010/01 schema in the document (section 5.4) is accurate and complete. It does not include any of the supplementary elements or attributes, as represented either in the normative textual descriptions of particular elements or in the appendix.

Further, the full supplementary schemas for micro-versioned RDL 2011/01, RDL 2012/01, and RDL 2013/01 that are found in sections 5.5, 5.6, and 5.7, respectively, are to be used in conjunction with RDL 2010/01, and examples of the XSD for the supplementary schemas are not included in the normative textual description of the particular element being described.

As an XML file that conforms to an XML Schema, RDL is comprised of simple and complex elements. RDL has other structural constraints and restrictions, as well, which cannot be expressed fully by an XML Schema.

Two facts are critical to the interpretation of the XML Schema definitions (XSDs) that are used throughout this document:

- Because of limitations in the XSD language, in some cases, the XSD does not fully constrain the contents of RDL. Some constraints are defined only in the normative textual description of the element.
- The XSD example, when taken out of context, can, in some cases, appear to be less restrictive than the normative textual description of the element, even when the full XSD actually matches the normative textual description exactly.

As an example of the second issue, consider the following XSD.

```
<xsd:element name="Child1" type="xsd:string" />
```

Based on this line, it would appear that **Child1** is mandatory.

However, the full context of the usage of **Child1** might be the following.

```
<xsd:complexType name="Parent">  
  <xsd:choice minOccurs="1" maxOccurs="unbounded">  
    <xsd:element name="Child1" type="xsd:string" />  
    <xsd:element name="Child2" type="xsd:string" />  
  </xsd:choice>  
</xsd:complexType>
```

The **xsd:choice** requires **Parent** to have at least one child, but **Child1** itself is not actually mandatory.

## 2.2 Common RDL Types

### 2.2.1 String

If the RDL format specifies an element or attribute as type **String**, it MUST be a valid **UNICODE** string, as specified in [\[XMLSCHEMA2/2\]](#) section 3.2.1 for **xsd:string**. Unless the value is explicitly specified (such as when the element or attribute is not present), its value is interpreted as a **null value**.

### 2.2.2 Integer

If the RDL format specifies an element or attribute as type **Integer**, it MUST be a valid **int** value, as specified in [\[XMLSCHEMA2/2\]](#) section 3.3.17 for **xsd:int**. A valid **int** value MUST be greater than -2147483649 and MUST be less than 2147483648.

### 2.2.3 Boolean

If the RDL format specifies an element or attribute as type **Boolean**, it MUST be a valid **Boolean** value, as specified in [\[XMLSCHEMA2/2\]](#) section 3.2.2 for **xsd:Boolean**.

## 2.2.4 Float

If the RDL format specifies an element or attribute as type **Float**, it MUST be a valid IEEE single-precision 32-bit floating point type [\[IEEE754\]](#), as specified in [\[XMLSCHEMA2\]](#) section 3.2.4 for **xsd:float**.

## 2.2.5 DateTime

If the RDL format specifies an element or attribute as type **DateTime**, it MUST be a valid date and time-of-day value as specified in [\[ISO8601\]](#) and in [\[XMLSCHEMA2\]](#) section 3.2.7 for **xsd:datetime**.

## 2.2.6 NormalizedString Elements and Attributes

The RDL format specifies a number of elements and attributes in the XSD structure as **xsd:NormalizedString**. This enforces further restrictions on the identifier string values for these elements and attributes.

Unless specified otherwise, when the value of an element or attribute is **xsd:NormalizedString**, in the file format validation implementation of any RDL schema, the **xsd:NormalizedString** identifiers MUST be the following:

- Unique within the entire **report**.
- **CLS-compliant identifier** [\[UTR15\]](#) name.

Further, [RDL 2003/10](#) and [RDL 2005/01](#), do not restrict the maximum length of identifiers, although [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#) restrict the maximum length of identifiers to 256 characters.

Any element whose value is **xsd:NormalizedString** can be referenced by the following elements and attributes.

Referenced by
<a href="#">RdlSize</a>
<a href="#">Image</a>
<a href="#">Line</a>
<a href="#">Textbox</a>
<a href="#">Rectangle</a>
<a href="#">Subreport</a>
<a href="#">Grouping</a>
<a href="#">Grouping.Name</a>
<a href="#">List</a>
<a href="#">List.Name</a>
<a href="#">Matrix</a>
<a href="#">Matrix.Name</a>
<a href="#">Table</a>
<a href="#">Table.Name</a>



<b>Referenced by</b>
<a href="#">Chart</a>
<a href="#">ChartArea</a>
<a href="#">ChartAxis</a>
<a href="#">ChartCodeParameter</a>
<a href="#">ChartLegend</a>
<a href="#">ChartLegendColumn</a>
<a href="#">ChartLegendCustomItem</a>
<a href="#">ChartLegendCustomItemCell</a>
<a href="#">ChartSeries</a>
<a href="#">ChartTitle</a>
<a href="#">CustomReportItem</a>
<a href="#">CustomReportItem.Name</a>
<a href="#">CustomReportItem.Type</a>
<a href="#">DataSet</a>
<a href="#">Field</a>
<a href="#">Group</a>
<a href="#">Group.Name</a>
<a href="#">Tablix</a>
<a href="#">Tablix.Name</a>
<a href="#">CustomLabel</a>
<a href="#">GaugeImage</a>
<a href="#">GaugeLabel</a>
<a href="#">GaugePanel</a>
<a href="#">IndicatorState</a>
<a href="#">LinearGauge</a>
<a href="#">LinearPointer</a>
<a href="#">LinearScale</a>
<a href="#">NumericIndicator</a>
<a href="#">NumericIndicatorRange</a>
<a href="#">RadialGauge</a>
<a href="#">RadialPointer</a>
<a href="#">RadialScale</a>

Referenced by
<a href="#">ScaleRange</a>
<a href="#">StateIndicator</a>
<a href="#">Class</a>
<a href="#">Class.InstanceName</a>
<a href="#">EmbeddedImage</a>
<a href="#">Variable</a>
<a href="#">ReportParameter</a>

## 2.2.7 RdISize

This simple type specifies restrictions for a valid size string value or expression.

Elements of the **RdISize** type MUST be in the following structure:

```
"<FormatValue><FormatType>"
```

The **FormatValue** property MUST have a numerical string value in the form of an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) with no thousand delimiters, optionally followed by a decimal point and a positive integer. Unless otherwise specified, the value of the **FormatValue** property MUST NOT evaluate to a negative **Integer** or [Float](#) ([XMLSCHEMA2] section 3.2.4) string.

The **FormatType** property specifies the unit type that is used for the distance. [<2>](#) This property MUST be set to one of the following:

**pt**: Specifies a size in units of points.

**pc**: Specifies a size in units of picas.

**in**: Specifies a size in units of inches.

**mm**: Specifies a size in units of millimeters.

**cm**: Specifies a size in units of centimeters.

Unless otherwise specified, if an element that is of type **RdISize** is not present, the value of the element is interpreted as

```
"0<FormatType>"
```

where *<FormatType>* is defined above.

A negative **RdISize** is an **RdISize** in the format of *<value><unit>* where *value* is a negative value.

The **RdISize** type can be referenced by the following elements.

Referenced by
<a href="#">Border.Width</a>

<b>Referenced by</b>
<a href="#">CapImage.OffsetX</a>
<a href="#">CapImage.OffsetY</a>
<a href="#">Image.Height</a>
<a href="#">Image.Left</a>
<a href="#">Image.Top</a>
<a href="#">Image.Width</a>
<a href="#">Line.Height</a>
<a href="#">Line.Left</a>
<a href="#">Line.Top</a>
<a href="#">Line.Width</a>
<a href="#">Page.BottomMargin</a>
<a href="#">Page.ColumnSpacing</a>
<a href="#">Page.InteractiveHeight</a>
<a href="#">Page.InteractiveWidth</a>
<a href="#">Page.LeftMargin</a>
<a href="#">Page.PageHeight</a>
<a href="#">Page.PageWidth</a>
<a href="#">Page.RightMargin</a>
<a href="#">PageSection.Height</a>
<a href="#">PointerImage.OffsetX</a>
<a href="#">PointerImage.OffsetY</a>
<a href="#">Rectangle.Height</a>
<a href="#">Rectangle.Left</a>
<a href="#">Rectangle.Top</a>
<a href="#">Rectangle.Width</a>
<a href="#">Style.LineHeight</a>
<a href="#">Style.PaddingBottom</a>
<a href="#">Style.PaddingLeft</a>
<a href="#">Style.PaddingRight</a>
<a href="#">Style.PaddingTop</a>
<a href="#">Textbox.Height</a>
<a href="#">Textbox.Left</a>

Referenced by
<a href="#">Textbox.Top</a>
<a href="#">Textbox.Width</a>
<a href="#">Paragraph.HangingIndent</a>
<a href="#">Paragraph.RightIndent</a>
<a href="#">Paragraph.SpaceAfter</a>
<a href="#">Paragraph.SpaceBefore</a>
<a href="#">Paragraph.LeftIndent</a>
<a href="#">Style.FontSize</a>
<a href="#">Style.ShadowOffset</a>
<a href="#">GaugeLabel.TextShadowOffset</a>
<a href="#">GaugePanel.Height</a>
<a href="#">GaugePanel.Left</a>
<a href="#">GaugePanel.Top</a>
<a href="#">GaugePanel.Width</a>

The following is the XML Schema definition of the **RdlSize** type.

```
<xsd:simpleType name="SizeType">
  <xsd:restriction base="xsd:normalizedString">
    </xsd:restriction>
  </xsd:simpleType>
```

### 2.2.8 RdlColor

The **RdlColor** type specifies a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) value or an expression that evaluates to a **String** where any of the following is true:

- Evaluates to be equal to one of the known color values that are listed in this section.
- Starts with a "#" (pound) character followed by a 6-digit hexadecimal expression.
- Starts with a "#" (pound) character followed by an 8-digit hexadecimal expression.

The **RdlColor** type can be referenced by the following elements.

Referenced by
<a href="#">BackgroundImage.TransparentColor</a>
<a href="#">Border.Color</a>
<a href="#">CapImage.HueColor</a>
<a href="#">CapImage.TransparentColor</a>
<a href="#">FrameImage.HueColor</a>

Referenced by
<a href="#">FrameImage.TransparentColor</a>
<a href="#">GaugeImage.TransparentColor</a>
<a href="#">MapMarkerImage.TransparentColor</a>
<a href="#">PointerImage.HueColor</a>
<a href="#">PointerImage.TransparentColor</a>
<a href="#">ScaleRange.InRangeBarPointerColor</a>
<a href="#">ScaleRange.InRangeTickMarksColor</a>
<a href="#">StateImage.TransparentColor</a>
<a href="#">Style.BackgroundColor</a>
<a href="#">Style.BackgroundGradientEndColor</a>
<a href="#">Style.Color</a>
<a href="#">Style.ShadowColor</a>
<a href="#">TopImage.HueColor</a>
<a href="#">TopImage.TransparentColor</a>

Following are the known color values for the **RdIColor** type.

Color name
AliceBlue
AntiqueWhite
Aqua
Aquamarine
Azure
Beige
Bisque
Black
BlanchedAlmond
Blue
BlueViolet
Brown
BurlyWood
CadetBlue
Chartreuse

<b>Color name</b>
Chocolate
Coral
CornflowerBlue
Cornsilk
Crimson
Cyan
DarkBlue
DarkCyan
DarkGoldenrod
DarkGray
DarkGreen
DarkKhaki
DarkMagenta
DarkOliveGreen
DarkOrange
DarkOrchid
DarkRed
DarkSalmon
DarkSeaGreen
DarkSlateBlue
DarkSlateGray
DarkTurquoise
DarkViolet
DeepPink
DeepSkyBlue
DimGray
DodgerBlue
Firebrick
FloralWhite
ForestGreen
Fuchsia
Gainsboro

<b>Color name</b>
GhostWhite
Gold
Goldenrod
Gray
Green
GreenYellow
Honeydew
HotPink
IndianRed
Indigo
Ivory
Khaki
Lavender
LavenderBlush
LawnGreen
LemonChiffon
LightBlue
LightCoral
LightCyan
LightGoldenrodYellow
LightGray
LightGreen
LightPink
LightSalmon
LightSeaGreen
LightSkyBlue
LightSlateGray
LightSteelBlue
LightYellow
Lime
LimeGreen
Linen

<b>Color name</b>
Magenta
Maroon
MediumAquaMarine
MediumBlue
MediumOrchid
MediumPurple
MediumSeaGreen
MediumSlateBlue
MediumSpringGreen
MediumTurquoise
MediumVioletRed
MidnightBlue
MintCream
MistyRose
Moccasin
NavajoWhite
Navy
OldLace
Olive
OliveDrab
Orange
OrangeRed
Orchid
PaleGoldenrod
PaleGreen
PaleTurquoise
PaleVioletRed
PapayaWhip
PeachPuff
Peru
Pink
Plum



Color name
PowderBlue
Purple
Red
RosyBrown
RoyalBlue
SaddleBrown
Salmon
SandyBrown
SeaGreen
SeaShell
Sienna
Silver
SkyBlue
SlateBlue
SlateGray
Snow
SpringGreen
SteelBlue
Tan
Teal
Thistle
Tomato
Turquoise
Violet
Wheat
White
WhiteSmoke
Yellow
YellowGreen
ButtonFace
ButtonHighlight
ButtonShadow

Color name
GradientActiveCaption
GradientInactiveCaption
MenuBar
MenuHighlight

### 2.2.9 RdlURL

The **RdlURL** type specifies a valid absolute URL. The value of this element MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The URL string or expression MUST begin with one of the following:

- http://
- https://
- ftp://
- mailto:
- news:

The **RdlURL** type can be referenced by the following elements.

Referenced by
<a href="#">Action.Hyperlink</a>
<a href="#">BackgroundImage.Value</a>
<a href="#">CapImage.Value</a>
<a href="#">FrameImage.Value</a>
<a href="#">GaugeImage.Value</a>
<a href="#">Image.Value</a>
<a href="#">MapMarkerImage.Value</a>
<a href="#">PointerImage.Value</a>
<a href="#">StateImage.Value</a>
<a href="#">TopImage.Value</a>

### 2.2.10 ReportMIMETYPE

The **ReportMIMETYPE** type specifies a [String](#) ([XMLSCHEMA2/2] section 3.2.1) value or expression that MUST evaluate to one of the following string values:

**image/bmp**

**image/jpeg**

**image/gif**

**image/png**

**image/x-png**

The **ReportMIMETYPE** type can be referenced by the following elements.

Referenced by
<a href="#">BackgroundImage.MIMETYPE</a>
<a href="#">CapImage.MIMETYPE</a>
<a href="#">EmbeddedImage.MIMETYPE</a>
<a href="#">FrameImage.MIMETYPE</a>
<a href="#">GaugeImage.MIMETYPE</a>
<a href="#">Image.MIMETYPE</a>
<a href="#">MapMarkerImage.MIMETYPE</a>
<a href="#">PointerImage.MIMETYPE</a>
<a href="#">StateImage.MIMETYPE</a>
<a href="#">TopImage.MIMETYPE</a>

### 2.2.11 ReportPath

The **ReportPath** type specifies either an **absolute path** or a **relative path** to a resource or an item to be used within a [Report](#). The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The **ReportPath** type can be referenced by the following elements.

Referenced by
<a href="#">BackgroundImage.Value</a>
<a href="#">CapImage.Value</a>
<a href="#">DataSource.DataSourceReference</a>
<a href="#">Drillthrough.ReportName</a>
<a href="#">FrameImage.Value</a>
<a href="#">GaugeImage.Value</a>
<a href="#">Image.Value</a>
<a href="#">MapMarkerImage.Value</a>
<a href="#">PointerImage.Value</a>

Referenced by
<a href="#">Report.DataTransform</a>
<a href="#">StateImage.Value</a>
<a href="#">Subreport.ReportName</a>
<a href="#">TopImage.Value</a>

### 2.2.12 ReportLanguage

The **ReportLanguage** type specifies a [String](#) ([XMLSCHEMA2/2] section 3.2.1) value or expression that MUST be a valid language code. The language code is a combination of the following:

- A two-letter lowercase culture code that is associated with a language, as specified in [\[ISO639-2\]](#)
- Optionally followed by a hyphen and a two-letter uppercase subculture code that is associated with a country or region, as specified in [\[ISO3166-1\]](#)

The **ReportLanguage** type can be referenced by the following elements.

Referenced by
<a href="#">Report.Language</a>
<a href="#">Style.Language</a>
<a href="#">Style.NumeralLanguage</a>
<a href="#">DataSet.CollationCulture</a>

### 2.2.13 Complex Types

If the RDL format specifies an optional RDL element as a complex type ([XMLSCHEMA1/2] section 3.4) and that complex type represents a sequence collection (of children elements of the same type), unless otherwise explicitly specified for that particular element, omitting that optional RDL element specifies an empty collection.

### 2.2.14 StringWithDataModelAttribute

The **StringWithDataModelAttribute** type specifies a [String](#) ([XMLSCHEMA2/2] section 3.2.1) value or expression and a **DataAttribute** attribute. The **DataAttribute** attribute specifies the data type of a value if the value is a constant. This attribute is optional. If this attribute is present, its value MUST be a **String**. The attribute **String** value MUST be the name of one of the following RDL data types: **String**, [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2), [DateTime](#) ([XMLSCHEMA2] section 3.2.7), [Integer](#) ([XMLSCHEMA2/2] section 3.3.17), or [Float](#) ([XMLSCHEMA2] section 3.2.4). If **DataAttribute** attribute is not present, the value of the **StringWithDataModelAttribute** type is interpreted as a **String**.

The **StringWithDataModelAttribute** type can be referenced by the following elements.

Referenced by
<a href="#">Field.Value</a>

<b>Referenced by</b>
<a href="#">FilterValues.FilterValue</a>
<a href="#">QueryParameter.Value</a>
<a href="#">Variable.Value</a>

The following is the XML Schema definition of the **StringWithDataModelAttribute** type in [RDL 2008/01](#).

```
<xsd:complexType name="StringWithDataModelAttribute">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="DataType" use="optional">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Boolean" />
            <xsd:enumeration value="DateTime" />
            <xsd:enumeration value="Integer" />
            <xsd:enumeration value="Float" />
            <xsd:enumeration value="String" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:anyAttribute namespace="##other" processContents="skip" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

The following is the XML Schema definition of the **StringWithDataModelAttribute** type in [RDL 2010/01](#) and [RDL 2016/01](#).

```
<xsd:complexType name="StringWithDataModelAttribute">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="DataType" use="optional">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Boolean" />
            <xsd:enumeration value="DateTime" />
            <xsd:enumeration value="Integer" />
            <xsd:enumeration value="Float" />
            <xsd:enumeration value="String" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

### 2.2.15 LocIDStringWithDataModelAttribute

The **LocIDStringWithDataModelAttribute** type specifies a [StringWithDataModelAttribute](#) with an **EvaluationMode** attribute. The **EvaluationMode** attribute specifies whether the value of the **LocIDStringWithDataModelAttribute** type is treated as an expression or constant. The **EvaluationMode** attribute is optional. If this attribute is present, its value MUST be one of the following:

**Auto:** Specifies that value MUST be treated as an expression if it starts with "=" and as a constant otherwise.

**Constant:** Specifies that value MUST be treated as a constant.

If the **EvaluationMode** attribute is not present, its value is interpreted as "Auto".

The **LocIDStringWithDataModelAttribute** type can be referenced by the following elements.

Referenced by
<a href="#">TextRun.Value</a>

The following is the XML Schema definition of the **LocIDStringWithDataModelAttribute** type in [RDL 2008/01](#).

```
<xsd:complexType name="LocIDStringWithDataModelAttribute">
  <xsd:simpleContent>
    <xsd:extension base="StringWithDataModelAttribute">
      <xsd:attribute name="EvaluationMode" type="EvaluationModeType" default="Auto" />
      <xsd:anyAttribute namespace="##other" processContents="skip" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="EvaluationModeType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Auto" />
    <xsd:enumeration value="Constant" />
  </xsd:restriction>
</xsd:simpleType>
```

The following is the XML Schema definition of the **LocIDStringWithDataModelAttribute** type in [RDL 2010/01](#) and [RDL 2016/01](#).

```
<xsd:complexType name="LocIDStringWithDataModelAttribute">
  <xsd:simpleContent>
    <xsd:extension base="StringWithDataModelAttribute">
      <xsd:attribute name="EvaluationMode" type="EvaluationModeType" default="Auto" />
      <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="EvaluationModeType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Auto" />
    <xsd:enumeration value="Constant" />
  </xsd:restriction>
</xsd:simpleType>
```

## 2.2.16 StringWithValueModelAttribute

**Applies to** [RDL 2012/01](#)

The **StringWithValueModelAttribute** type specifies a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) value or expression and a **ValueType** attribute. The **ValueType** attribute specifies whether the value of the **StringWithValueModelAttribute** type is treated as a theme reference if the value is not an expression. The **ValueType** attribute is optional. If this attribute is present, its value MUST be one of the following:

**Constant:** Specifies that the value is to be interpreted as a constant.

**ThemeReference:** Specifies that the value refers to a value specified in a theme.

If a **ValueType** attribute is not present, the value of the **StringWithValueTypeAttribute** type is interpreted as "Constant".

If a **ValueType** attribute is not present, its value is interpreted as "Constant".

The **StringWithValueTypeAttribute** type can be referenced by the following elements.

Referenced by
<a href="#">Style.BackgroundColor</a>
<a href="#">Style.Color</a>
<a href="#">Style.FontFamily</a>

The following is the XML Schema definition of the **StringWithValueTypeAttribute** type.

```
<xsd:complexType name="StringWithValueTypeAttribute">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute ref="ValueType" use="optional" />
      <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

## 2.3 Report

The **Report** element specifies the structure, data, and layout information of a **Report**. The following elements MUST be specified for a **Report** element:

- [Report.Body](#) and [Report.Width](#) in [RDL 2003/10](#) and [RDL 2005/01](#).
- **Report.Body**, [Report.Page](#), and **Report.Width** in [RDL 2008/01](#).
- [Report.ReportSections](#) in [RDL 2010/01](#) and [RDL 2016/01](#).

The following element can be specified for a **Report** element:

- [Report.ReportParametersLayout](#) in RDL 2016/01.

The following are the child attributes of the **Report** element.

Child attributes
<a href="#">Report.MustUnderstand</a>

The following are the child elements of the **Report** element.

Child elements
<a href="#">Report.AuthoringMetadata</a>
<a href="#">Report.Author</a>
<a href="#">Report.AutoRefresh</a>
Report.Body

<b>Child elements</b>
<a href="#">Report.Classes</a>
<a href="#">Report.Code</a>
<a href="#">Report.CodeModules</a>
<a href="#">Report.ConsumeContainerWhiteSpace</a>
<a href="#">Report.CustomProperties</a>
<a href="#">Report.DataElementName</a>
<a href="#">Report.DataElementStyle</a>
<a href="#">Report.DataSchema</a>
<a href="#">Report.DataSets</a>
<a href="#">Report.DataSources</a>
<a href="#">Report.DataTransform</a>
<a href="#">Report.DefaultFontFamily</a>
<a href="#">Report.DeferVariableEvaluation</a>
<a href="#">Report.Description</a>
<a href="#">Report.EmbeddedImages</a>
<a href="#">Report.InitialPageName</a>
<a href="#">Report.InteractiveHeight</a>
<a href="#">Report.InteractiveWidth</a>
<a href="#">Report.Language</a>
<a href="#">Report.BottomMargin</a>
<a href="#">Report.LeftMargin</a>
<a href="#">Report.TopMargin</a>
<a href="#">Report.RightMargin</a>
<a href="#">Report.PageFooter</a>
<a href="#">Report.PageHeader</a>
<a href="#">Report.PageHeight</a>
<a href="#">Report.PageWidth</a>
Report.Page
<a href="#">Report.ReportParameters</a>
Report.ReportParametersLayout
<a href="#">Report.Variables</a>
Report.Width



**Child elements**

Report.ReportSections

The following is the XML Schema definition of the **Report** element in RDL 2003/10.

```
<xsd:element name="Report">
  <xsd:complexType>
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="Description" type="StringLocIDType" minOccurs="0" />
      <xsd:element name="Author" type="xsd:string" minOccurs="0" />
      <xsd:element name="AutoRefresh" type="xsd:unsignedInt" minOccurs="0" />
      <xsd:element name="DataSources" type="DataSourcesType" minOccurs="0" />
      <xsd:element name="DataSets" type="DataSetsType" minOccurs="0" />
      <xsd:element name="ReportParameters" type="ReportParametersType" minOccurs="0" />
      <xsd:element name="Code" type="xsd:string" minOccurs="0" />
      <xsd:element name="EmbeddedImages" type="EmbeddedImagesType" minOccurs="0" />
      <xsd:element name="Language" type="xsd:string" minOccurs="0" />
      <xsd:element name="CodeModules" type="CodeModulesType" minOccurs="0" />
      <xsd:element name="Classes" type="ClassesType" minOccurs="0" />
      <xsd:element name="Custom" type="Custom" minOccurs="0" />
      <xsd:element name="DataTransform" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataSchema" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataElementStyle" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="AttributeNormal" />
            <xsd:enumeration value="ElementNormal" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="PageHeader" type="PageHeaderFooterType" minOccurs="0" />
      <xsd:element name="PageFooter" type="PageHeaderFooter" minOccurs="0" />
      <xsd:element name="PageHeight" type="SizeType" minOccurs="0" />
      <xsd:element name="PageWidth" type="SizeType" minOccurs="0" />
      <xsd:element name="InteractiveHeight" type="SizeType" minOccurs="0" />
      <xsd:element name="InteractiveWidth" type="SizeType" minOccurs="0" />
      <xsd:element name="LeftMargin" type="SizeType" minOccurs="0" />
      <xsd:element name="RightMargin" type="SizeType" minOccurs="0" />
      <xsd:element name="TopMargin" type="SizeType" minOccurs="0" />
      <xsd:element name="BottomMargin" type="SizeType" minOccurs="0" />
      <xsd:element name="Body" type="BodyType" minOccurs="1" />
      <xsd:element name="Width" type="SizeType" minOccurs="1" />
      <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>
</xsd:element>
```

The following is the XML Schema definition of the **Report** element in RDL 2005/01.

```
<xsd:element name="Report">
  <xsd:complexType>
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="Description" type="xsd:string" minOccurs="0" />
      <xsd:element name="Author" type="xsd:string" minOccurs="0" />
      <xsd:element name="AutoRefresh" type="xsd:unsignedInt" minOccurs="0" />
      <xsd:element name="DataSources" type="DataSourcesType" minOccurs="0" />
      <xsd:element name="DataSets" type="DataSetsType" minOccurs="0" />
      <xsd:element name="Body" type="BodyType" />
      <xsd:element name="ReportParameters" type="ReportParametersType" minOccurs="0" />
      <xsd:element name="Code" type="xsd:string" minOccurs="0" />
      <xsd:element name="Width" type="SizeType" />
      <xsd:element name="PageHeader" type="PageHeaderFooterType" minOccurs="0" />
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

```

<xsd:element name="PageFooter" type="PageHeaderFooterType" minOccurs="0" />
<xsd:element name="PageHeight" type="SizeType" minOccurs="0" />
<xsd:element name="PageWidth" type="SizeType" minOccurs="0" />
<xsd:element name="InteractiveHeight" type="SizeType" minOccurs="0" />
<xsd:element name="InteractiveWidth" type="SizeType" minOccurs="0" />
<xsd:element name="LeftMargin" type="SizeType" minOccurs="0" />
<xsd:element name="RightMargin" type="SizeType" minOccurs="0" />
<xsd:element name="TopMargin" type="SizeType" minOccurs="0" />
<xsd:element name="BottomMargin" type="SizeType" minOccurs="0" />
<xsd:element name="EmbeddedImages" type="EmbeddedImagesType" minOccurs="0" />
<xsd:element name="Language" type="xsd:string" minOccurs="0" />
<xsd:element name="CodeModules" type="CodeModulesType" minOccurs="0" />
<xsd:element name="Classes" type="ClassesType" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
<xsd:element name="DataTransform" type="xsd:string" minOccurs="0" />
<xsd:element name="DataSchema" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementStyle" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="AttributeNormal" />
      <xsd:enumeration value="ElementNormal" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
</xsd:element>

```

The following is the XML Schema definition of the **Report** element in RDL 2008/01.

```

<xsd:element name="Report">
  <xsd:complexType>
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="Description" type="StringLocIDType" minOccurs="0" />
      <xsd:element name="Author" type="xsd:string" minOccurs="0" />
      <xsd:element name="AutoRefresh" type="xsd:unsignedInt" minOccurs="0" />
      <xsd:element name="DataSources" type="DataSourcesType" minOccurs="0" />
      <xsd:element name="DataSets" type="DataSetsType" minOccurs="0" />
      <xsd:element name="ReportParameters" type="ReportParametersType" minOccurs="0" />
      <xsd:element name="Code" type="xsd:string" minOccurs="0" />
      <xsd:element name="EmbeddedImages" type="EmbeddedImagesType" minOccurs="0" />
      <xsd:element name="Language" type="xsd:string" minOccurs="0" />
      <xsd:element name="CodeModules" type="CodeModulesType" minOccurs="0" />
      <xsd:element name="Classes" type="ClassesType" minOccurs="0" />
      <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
      <xsd:element name="DataTransform" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataSchema" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataElementStyle" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Attribute" />
            <xsd:enumeration value="Element" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="Variables" type="VariablesType" minOccurs="0" />
      <xsd:element name="DeferVariableEvaluation" type="xsd:boolean" minOccurs="0" />
      <xsd:element name="ConsumeContainerWhitespace" type="xsd:boolean" minOccurs="0" />
      <xsd:element name="Page" type="PageType" minOccurs="1" />
      <xsd:element name="Body" type="BodyType" minOccurs="1" />
      <xsd:element name="Width" type="SizeType" minOccurs="1" />
      <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

```

```

    <xsd:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>
</xsd:element>

```

The following is the XML Schema definition of the **Report** element in RDL 2010/01.

```

<xsd:element name="Report">
  <xsd:complexType>
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="Description" type="StringLocIDType" minOccurs="0" />
      <xsd:element name="Author" type="xsd:string" minOccurs="0" />
      <xsd:element name="AutoRefresh" type="xsd:string" minOccurs="0" />
      <xsd:element name="InitialPageName" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataSources" type="DataSourcesType" minOccurs="0" />
      <xsd:element name="DataSets" type="DataSetsType" minOccurs="0" />
      <xsd:element name="ReportParameters" type="ReportParametersType" minOccurs="0" />
      <xsd:element name="Code" type="xsd:string" minOccurs="0" />
      <xsd:element name="EmbeddedImages" type="EmbeddedImagesType" minOccurs="0" />
      <xsd:element name="Language" type="xsd:string" minOccurs="0" />
      <xsd:element name="CodeModules" type="CodeModulesType" minOccurs="0" />
      <xsd:element name="Classes" type="ClassesType" minOccurs="0" />
      <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
      <xsd:element name="Variables" type="VariablesType" minOccurs="0" />
      <xsd:element name="DeferVariableEvaluation" type="xsd:boolean" minOccurs="0" />
      <xsd:element name="ConsumeContainerWhitespace" type="xsd:boolean" minOccurs="0" />
      <xsd:element name="DataTransform" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataSchema" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataElementStyle" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Attribute" />
            <xsd:enumeration value="Element" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="ReportSections" type="ReportSectionsType" minOccurs="1" />
      <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
</xsd:element>

```

The following is the XML Schema definition of the **Report** element in RDL 2016/01.

```

<xsd:element name="Report">
  <xsd:complexType>
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="DefaultFontFamily" type="xsd:string" minOccurs="0"
        xmlns="http://schemas.microsoft.com/sqlserver/reporting/2016/01/reportdefinition/defaultfontfamily" />
      <xsd:element name="AuthoringMetadata" type="xsd:string" minOccurs="0"
        xmlns="http://schemas.microsoft.com/sqlserver/reporting/2016/01/reportdefinition/authoringmetadata" />
      <xsd:element name="Description" type="StringLocIDType" minOccurs="0" />
      <xsd:element name="Author" type="xsd:string" minOccurs="0" />
      <xsd:element name="AutoRefresh" type="xsd:string" minOccurs="0" />
      <xsd:element name="InitialPageName" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataSources" type="DataSourcesType" minOccurs="0" />
      <xsd:element name="DataSets" type="DataSetsType" minOccurs="0" />
      <xsd:element name="ReportParameters" type="ReportParametersType" minOccurs="0" />
      <xsd:element name="ReportParametersLayout" type="ReportParametersLayoutType"
        minOccurs="0" />
      <xsd:element name="Code" type="xsd:string" minOccurs="0" />
      <xsd:element name="EmbeddedImages" type="EmbeddedImagesType" minOccurs="0" />
    </xsd:choice>
  </xsd:complexType>
</xsd:element>

```

```

<xsd:element name="Language" type="xsd:string" minOccurs="0" />
<xsd:element name="CodeModules" type="CodeModulesType" minOccurs="0" />
<xsd:element name="Classes" type="ClassesType" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
<xsd:element name="Variables" type="VariablesType" minOccurs="0" />
<xsd:element name="DeferVariableEvaluation" type="xsd:boolean" minOccurs="0" />
<xsd:element name="ConsumeContainerWhitespace" type="xsd:boolean" minOccurs="0" />
<xsd:element name="DataTransform" type="xsd:string" minOccurs="0" />
<xsd:element name="DataSchema" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementStyle" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Attribute" />
      <xsd:enumeration value="Element" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="ReportSections" type="ReportSectionsType" minOccurs="1" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="MustUnderstand" type="MustUnderstandType" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
</xsd:element>

```

### 2.3.1 Report.MustUnderstand

#### *Applies to* [RDL 2016/01](#)

The **Report.MustUnderstand** attribute specifies a whitespace delimited list of XML prefixes that reference XML namespaces that the report server MUST be able to process in order for the RDL document to be read. If this element is present, its value MUST be a whitespace delimited list ([\[XMLSCHEMA2\]](#) section 2.5.1.2) of tokens ([\[XMLSCHEMA2\]](#) section 3.3.2).

The following is the parent element of the **Report.MustUnderstand** attribute.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.MustUnderstand** attribute.

```

<xsd:attribute name="MustUnderstand" type="MustUnderstandType">
  <xsd:simpleType>
    <xsd:list itemType="xsd:token" />
  </xsd:simpleType>
</xsd:attribute>

```

### 2.3.2 Report.AuthoringMetadata

#### *Applies to* [RDL 2016/01](#)

The **Report.AuthoringMetadata** element allows authoring information about the tools that are used to create and edit a given rdl file. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

**Report.AuthoringMetadata** is purely metadata and has no functional implications.

The following is the parent element of the **Report.AuthoringMetadata** element.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.AuthoringMetadata** element.

```
<xsd:element name="AuthoringMetadata" type="xsd:string" minOccurs="0"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2016/01/reportdefinition/authoringmet
adata" />
```

### 2.3.3 Report.Author

The **Report.Author** element specifies the name of the author of a [Report](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Following is the parent element of the **Report.Author** element.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.Author** element.

```
<xsd:element name="Author" type="xsd:string" minOccurs="0" />
```

### 2.3.4 Report.AutoRefresh

The **Report.AutoRefresh** element specifies the rate at which the **report Page** automatically refreshes. The **Report.AutoRefresh** element is optional.

If this element is present in [RDL 2003/10](#), [RDL 2005/01](#), or [RDL 2008/01](#), its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17). If this element is present in [RDL 2010/01](#) or [RDL 2016/01](#), its value MUST be either an **Integer** or an expression that evaluates to an **Integer**. The value MUST be greater than or equal to 0 and less than or equal to 2147483647. If this element is not present, its value is interpreted as 0. If this element has a value of 0, the report **page** does not automatically refresh.

Following is the parent element of the **Report.AutoRefresh** element.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.AutoRefresh** element in RDL 2003/10, RDL 2005/01, and RDL 2008/01.

```
<xsd:element name="AutoRefresh" type="xsd:unsignedInt" minOccurs="0" />
```

The following is the XML Schema definition of the **Report.AutoRefresh** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:element name="AutoRefresh" type="xsd:string" minOccurs="0" />
```

### 2.3.5 Report.Body

Applies to [RDL 2003/10](#), [RDL 2005/01](#), and [RDL 2008/01](#)

The **Report.Body** element describes how the [Body](#) of the report is structured and rendered. The **Report.Body** element MUST be specified. This element is of type **Body**.

Following is the parent element of the **Report.Body** element.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.Body** element.

```
<xsd:element name="Body" type="BodyType" minOccurs="1" />
```

### 2.3.6 Report.Classes

The **Report.Classes** element specifies **classes** to instantiate during report initialization. This element is optional. This element is of type [Classes](#).

Following is the parent element of the **Report.Classes** element.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.Classes** element.

```
<xsd:element name="Classes" type="ClassesType" minOccurs="0" />
```

### 2.3.7 Report.Code

The **Report.Code** element specifies the code definitions for custom functions to be used in an expression within a [Report.<3>](#)

The **Report.Code** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Following is the parent element of the **Report.Code** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.Code** element.

```
<xsd:element name="Code" type="xsd:string" minOccurs="0" />
```

### 2.3.8 Report.CodeModules

The **Report.CodeModules** element specifies the code modules to make available to a [Report](#) for use in an expression. This element is optional. This element is of type [CodeModules](#).

Following is the parent element of the **Report.CodeModules** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.CodeModules** element.

```
<xsd:element name="CodeModules" type="CodeModulesType" minOccurs="0" />
```

### 2.3.9 Report.ConsumeContainerWhitespace

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Report.ConsumeContainerWhitespace** element indicates that all white space in containers (such as [Body](#) and [Rectangle](#)) is consumed during report rendering when the container's contents grow rather than preserving the minimum white space between the contents and the bounds of the container.

The **Report.ConsumeContainerWhitespace** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Report.ConsumeContainerWhitespace** element.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.ConsumeContainerWhitespace** element.

```
<xsd:element name="ConsumeContainerWhitespace" type="xsd:boolean" minOccurs="0" />
```

### 2.3.10 Report.CustomProperties

The **Report.CustomProperties** element specifies custom information for a [Report](#) instance that will be handed to a report rendering component. This element is optional. This element is of type [CustomProperties](#).

Following is the parent element of the **Report.CustomProperties** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

In [RDL 2003/10](#), the equivalent element of **Report.CustomProperties** is **Report.Custom**, which is of type [Custom](#).

### 2.3.11 Report.DataElementName

The **Report.DataElementName** element specifies the name to use for the top-most element in a report. This element is optional. If this element is present, its value MUST be a **CLS-compliant identifier** [UTR15].

Following is the parent element of the **Report.DataElementName** element.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.3.12 Report.DataElementStyle

The **Report.DataElementStyle** element indicates whether **leaf level** values (such as **text box** values and **chart** data values) in a report render as elements or as attributes in a **data rendering**. This element is optional.

If the **Report.DataElementStyle** element is present, its value MUST be either "AttributeNormal" or "ElementNormal" in [RDL 2003/10](#) and [RDL 2005/01](#), and either "Attribute" or "Element" in [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#). If this element is not present, its value is interpreted as "Element" or "ElementNormal" depending on the RDL version.

Following is the parent element of the **Report.DataElementStyle** element.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.DataElementStyle** element in RDL 2003/10 and RDL 2005/01.

```
<xsd:element name="DataElementStyle" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="AttributeNormal" />
      <xsd:enumeration value="ElementNormal" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

The following is the XML Schema definition of the **Report.DataElementStyle** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

```
<xsd:element name="DataElementStyle" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Attribute" />
      <xsd:enumeration value="Element" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```



### 2.3.13 Report.DataSchema

The **Report.DataSchema** element specifies the schema or namespace to use for a **data rendering**. The **Report.DataSchema** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1).

Following is the parent element of the **Report.DataSchema** element.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.DataSchema** element.

```
<xsd:element name="DataSchema" type="xsd:string" minOccurs="0" />
```

### 2.3.14 Report.DataSets

The **Report.DataSets** element specifies the **datasets** to use for a [Report](#). This element is optional. This element is of type [DataSets](#).

Following is the parent element of the **Report.DataSets** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.DataSets** element.

```
<xsd:element name="DataSets" type="DataSetsType" minOccurs="0" />
```

### 2.3.15 Report.DataSources

The **Report.DataSources** specifies the data sources to use for a [Report](#). This element is optional. This element is of type [DataSources](#).

Following is the parent element of the **Report.DataSources** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.DataSources** element.

```
<xsd:element name="DataSources" type="DataSourcesType" minOccurs="0" />
```

### 2.3.16 Report.DataTransform

The **Report.DataTransform** element specifies the location to a transformation to apply to a **data rendering**. This can be an **absolute path** (such as /xsl/xfrm.xsl) or a **relative path** (such as xfrm.xsl).

The **Report.DataTransform** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1).

Following is the parent element of the **Report.DataTransform** element.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.DataTransform** element.

```
<xsd:element name="DataTransform" type="xsd:string" minOccurs="0" />
```

### 2.3.17 Report.DefaultFontFamily

**Applies to** [RDL 2016/01](#)

The **Report.DefaultFontFamily** element specifies the font family that is used for all text that is rendered when an explicit font is not defined in the [Style.FontFamily](#) element of any child report item.

The **Report.DefaultFontFamily** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1). If this element is not present, its value is interpreted as "Arial". If the value of this element is not a recognized font family, its value is interpreted as "Arial".

Following is the parent element of the **Report.DefaultFontFamily** element.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.DefaultFontFamily** element.

```
<xsd:element name="DefaultFontFamily" type="xsd:string" minOccurs="0"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2016/01/reportdefinition/defaultfontf
amily" />
```

### 2.3.18 Report.DeferVariableEvaluation

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Report.DeferVariableEvaluation** element specifies that [Variables](#) throughout a [Report](#) are not required to be evaluated at the beginning of report processing and can be evaluated on-demand based on the variable's usage.

The **Report.DeferVariableEvaluation** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Report.DeferVariableEvaluation** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.DeferVariableEvaluation** element.

```
<xsd:element name="DeferVariableEvaluation" type="xsd:boolean" minOccurs="0" />
```

### 2.3.19 Report.Description

The **Report.Description** element specifies the description of a [Report](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Following is the parent element of the **Report.Description** element.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.Description** element.

```
<xsd:element name="Description" type="xsd:string" minOccurs="0" />
```

### 2.3.20 Report.EmbeddedImages

The **Report.EmbeddedImages** element specifies the **images** that are embedded within a [Report](#). This element is optional. This element is of type [EmbeddedImages](#).

Following is the parent element of the **Report.EmbeddedImages** element.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.EmbeddedImages** element.

```
<xsd:element name="EmbeddedImages" type="EmbeddedImagesType" minOccurs="0" />
```

### 2.3.21 Report.InitialPageName

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Report.InitialPageName** element indicates the initial value to use as the name of the paginated **page** by the **renderers**. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Report.InitialPageName** element.

Parent elements
<a href="#">Report</a>

The following is the XML Schema definition of the **Report.InitialPageName** element.

```
<xsd:element name="InitialPageName" type="xsd:string" minOccurs="0" />
```

### 2.3.22 Report.InteractiveHeight

**Applies to** [RDL 2005/01](#)

The **Report.InteractiveHeight** element specifies the default height for rendering a [Report](#) by an **interactive renderer**. The **Report.InteractiveHeight** element is optional. If this element is present,

its value MUST be an [RdlSize](#). If the **Report.InteractiveHeight** element is not present, its value is interpreted as the value of the [Report.PageHeight](#) element.

Following is the parent element of the **Report.InteractiveHeight** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.InteractiveHeight** element.

```
<xsd:element name="InteractiveHeight" type="SizeType" minOccurs="0" />
```

### 2.3.23 Report.InteractiveWidth

**Applies to** [RDL 2005/01](#)

The **Report.InteractiveWidth** element specifies the default width for rendering a [Report](#) by an **interactive renderer**. The **Report.InteractiveWidth** element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as the value of [Report.PageWidth](#).

Following is the parent element of the **Report.InteractiveWidth** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.InteractiveWidth** element.

```
<xsd:element name="InteractiveWidth" type="SizeType" minOccurs="0" />
```

### 2.3.24 Report.Language

The **Report.Language** element specifies the primary language of the text in a [Report](#). This value is used as the default language for all language-dependent expressions in the report. The **Report.Language** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Report.Language** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.Language** element.

```
<xsd:element name="Language" type="xsd:string" minOccurs="0" />
```

### 2.3.25 Report.BottomMargin

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Report.BottomMargin** element specifies the width of the bottom margin of a [Report](#). The **Report.BottomMargin** element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as "0in".

Following is the parent element of the **Report.BottomMargin** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.BottomMargin** element.

```
<xsd:element name="BottomMargin" type="SizeType" minOccurs="0" />
```

### 2.3.26 Report.LeftMargin

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Report.LeftMargin** element specifies the width of the left margin of a [Report](#). The **Report.LeftMargin** element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as "0in".

Following is the parent element of the **Report.LeftMargin** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.LeftMargin** element.

```
<xsd:element name="LeftMargin" type="SizeType" minOccurs="0" />
```

### 2.3.27 Report.TopMargin

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Report.TopMargin** element specifies the height of the top margin of a [Report](#). The **Report.TopMargin** element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as "0in".

Following is the parent element of the **Report.TopMargin** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.TopMargin** element.

```
<xsd:element name="TopMargin" type="SizeType" minOccurs="0" />
```

### 2.3.28 Report.RightMargin

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Report.RightMargin** element specifies the width of the right margin of a [Report](#). The **Report.RightMargin** element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as "0in".

Following is the parent element of the **Report.RightMargin** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.RightMargin** element.

```
<xsd:element name="RightMargin" type="SizeType" minOccurs="0" />
```

### 2.3.29 Report.PageFooter

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Report.PageFooter** element specifies a footer that is rendered at the bottom of each rendered [Page](#) of a [Report](#). This element is optional. This element is of type [PageHeaderFooter](#).

Following is the parent element of the **Report.PageFooter** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.PageFooter** element.

```
<xsd:element name="PageFooter" type="PageHeaderFooterType" minOccurs="0" />
```

### 2.3.30 Report.PageHeader

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Report.PageHeader** element specifies a header that is rendered at the top of each rendered [Page](#) of a [Report](#). The **Report.PageHeader** element is optional. This element is of type [PageHeaderFooter](#).

Following is the parent element of the **Report.PageHeader** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.PageHeader** element.

```
<xsd:element name="PageHeader" type="PageHeaderFooterType" minOccurs="0" />
```

### 2.3.31 Report.PageHeight

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Report.PageHeight** element specifies the default height of each rendered [Page](#) of a [Report](#) in a **physical-page oriented renderer**. The **Report.PageHeight** element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as "11in".

Following is the parent element of the **Report.PageHeight** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.PageHeight** element.

```
<xsd:element name="PageHeight" type="SizeType" minOccurs="0" />
```

### 2.3.32 Report.PageWidth

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **Report.PageWidth** element specifies the default width for rendering each [Page](#) of a [Report](#) in a **physical-page oriented renderer**. The **Report.PageWidth** element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as "8.5in".

Following is the parent element of the **Report.PageWidth** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.PageWidth** element.

```
<xsd:element name="PageWidth" type="SizeType" minOccurs="0" />
```

### 2.3.33 Report.Page

**Applies to [RDL 2008/01](#)**

The **Report.Page** element specifies page layout information about a [Report](#). This element is of type [Page](#). This element MUST be specified.

Following is the parent element of the **Report.Page** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.Page** element.

```
<xsd:element name="Page" type="PageType" minOccurs="1" />
```

### 2.3.34 Report.ReportParameters

The **Report.ReportParameters** element specifies the **report parameters** for a [Report](#). This element is optional. This element is of type [ReportParameters](#).

Following is the parent element of the **Report.ReportParameters** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.ReportParameters** element.

```
<xsd:element name="ReportParameters" type="ReportParametersType" minOccurs="0" />
```

### 2.3.35 Report.ReportParametersLayout

**Applies to** [RDL 2016/01](#)

The **Report.ReportParametersLayout** element specifies element information of a report parameter layout for a [Report](#). This element **MUST** be specified if the **Report.ReportParameters** element is specified. This element is of type [ReportParametersLayout](#).

Following is the parent element of the **Report.ReportParametersLayout** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.ReportParametersLayout** element.

```
<xsd:element name="ReportParametersLayout" type="ReportParametersLayoutType" minOccurs="0" />
```

### 2.3.36 Report.Variables

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Report.Variables** element specifies the [Variables](#) defined for a [Report](#) as a whole. The **Report.Variables** element is optional. This element is of type **Variables**.

Following is the parent element of the **Report.Variables** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.Variables** element.

```
<xsd:element name="Variables" type="VariablesType" minOccurs="0" />
```

### 2.3.37 Report.Width

**Applies to** [RDL 2003/10](#), [RDL 2005/01](#), and [RDL 2008/01](#)

The **Report.Width** element specifies the width of a [Report](#). This element **MUST** be specified, and its value **MUST** be an [RdlSize](#).

Following is the parent element of the **Report.Width** element.



Parent elements
Report

The following is the XML Schema definition of the **Report.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="1" />
```

### 2.3.38 Report.ReportSections

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Report.ReportSections** element specifies the [ReportSection](#) elements of a [Report](#). This element MUST be specified, and it MUST contain at least one **ReportSection** element.

Following is the parent element of the **Report.ReportSections** element.

Parent elements
Report

The following is the XML Schema definition of the **Report.ReportSections** element.

```
<xsd:element name="ReportSections" type="ReportSectionsType" minOccurs="1" />
```

## 2.4 AuthoringMetadata

**Applies to** [RDL 2016/01](#)

The **AuthoringMetadata** element specifies the authoring information about the [Report](#). This element is optional. **AuthoringMetadata** is purely metadata and has no functional implications.

The following are the parent and child elements of the **AuthoringMetadata** element.

Parent elements
Report

Child elements
<a href="#">AuthoringMetadata.CreatedBy</a>
<a href="#">AuthoringMetadata.UpdatedBy</a>
<a href="#">AuthoringMetadata.Name</a>
<a href="#">AuthoringMetadata.Version</a>

The following is the XML Schema definition of the **AuthoringMetadata** element.

```
<xsd:element name="AuthoringMetadata" type="xsd:string" minOccurs="0"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2016/01/reportdefinition/authoringmet
adata" />
<xsd:element name="CreatedBy" type="xsd:string" minOccurs="0" />
```

```
<xsd:element name="UpdatedBy" type="xsd:string" minOccurs="0" />
<xsd:element name="Name" type="xsd:string" minOccurs="0" />
<xsd:element name="Version" type="xsd:string" minOccurs="0" />
</xsd:element>
```

### 2.4.1 AuthoringMetadata.CreatedBy

**Applies to** [RDL 2016/01](#)

The **AuthoringMetadata.CreatedBy** element specifies the creator of the report. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1).

The following is the parent element of the **AuthoringMetadata.CreatedBy** element.

Parent elements
<a href="#">AuthoringMetadata</a>

The following is the XML Schema definition of the **AuthoringMetadata.CreatedBy** element.

```
<xsd:element name="CreatedBy" type="xsd:string" minOccurs="0" />
```

### 2.4.2 AuthoringMetadata.UpdatedBy

**Applies to** [RDL 2016/01](#)

The **AuthoringMetadata.UpdatedBy** element specifies the date the report was last updated. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1).

The following is the parent element of the **AuthoringMetadata.UpdatedBy** element.

Parent elements
<a href="#">AuthoringMetadata</a>

The following is the XML Schema definition of the **AuthoringMetadata.UpdatedBy** element.

```
<xsd:element name="UpdatedBy" type="xsd:string" minOccurs="0" />
```

### 2.4.3 AuthoringMetadata.Name

**Applies to** [RDL 2016/01](#)

The **AuthoringMetadata.Name** element specifies the name of the report. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1).

The following is the parent element of the **AuthoringMetadata.Name** element.

<b>Parent elements</b>
------------------------

<a href="#">AuthoringMetadata</a>
-----------------------------------

The following is the XML Schema definition of the **AuthoringMetadata.Name** element.

```
<xsd:element name="Name" type="xsd:string" minOccurs="0" />
```

## 2.4.4 AuthoringMetadata.Version

*Applies to [RDL 2016/01](#)*

The **AuthoringMetadata.Version** element specifies the RDL schema version that was used to craft or edit the report. This element is optional. If this element is present, its value **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

The following is the parent element of the **AuthoringMetadata.Version** element.

<b>Parent elements</b>
------------------------

<a href="#">AuthoringMetadata</a>
-----------------------------------

The following is the XML Schema definition of the **AuthoringMetadata.Version** element.

```
<xsd:element name="Version" type="xsd:string" minOccurs="0" />
```

## 2.5 ReportSections

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **ReportSections** element specifies an ordered list of [ReportSection](#) elements, which comprise the structure and layout information for a [Report](#). The **ReportSections** element **MUST** be specified, and **MUST** contain at least one child **ReportSection**.

The following are the parent elements and child elements of the **ReportSections** element.

<b>Parent elements</b>
------------------------

Report
--------

<b>Child elements</b>
-----------------------

<a href="#">ReportSections.ReportSection</a>
--

The following is the XML Schema definition of the **ReportSections** element.

```
<xsd:complexType name="ReportSectionsType">
  <xsd:sequence>
    <xsd:element name="ReportSection" type="ReportSectionType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.5.1 ReportSections.ReportSection

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **ReportSections.ReportSection** element specifies the definition of a **report section**. This element MUST be specified. This element is of type [ReportSection](#).

Following is the parent element of the **ReportSections.ReportSection** element.

Parent elements
<a href="#">ReportSections</a>

The following is the XML Schema definition of the **ReportSections.ReportSection** element.

```
<xsd:element name="ReportSection" type="ReportSectionType" minOccurs="1"
maxOccurs="unbounded" />
```

## 2.6 ReportSection

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **ReportSection** element specifies the structure and layout information of a **ReportSection**. This element MUST be specified.

The following are the parent elements and child elements of the **ReportSection** element.

Parent elements
<a href="#">ReportSections</a>

Child elements
<a href="#">ReportSection.Body</a>
<a href="#">ReportSection.Page</a>
<a href="#">ReportSection.Width</a>
<a href="#">ReportSection.DataElementName</a>
<a href="#">ReportSection.DataElementOutput</a>

Applies to [RDL 2011/01](#)

Attributes
<a href="#">ReportSection.Name</a>

Applies to [RDL 2012/01](#)

Child elements
<a href="#">ReportSection.LayoutDirection</a>

The following is the XML Schema definition of the **ReportSection** element.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, 5.6, and 5.7, respectively. For more information about macro- and micro-versioned schemas, see section 2.1.

```
<xsd:complexType name="ReportSectionType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Body" type="BodyType" minOccurs="1" />
    <xsd:element name="Width" type="SizeType" minOccurs="1" />
    <xsd:element name="Page" type="PageType" minOccurs="1" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.6.1 ReportSection.Body

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **ReportSection.Body** element describes the structure and layout of the body of a [ReportSection](#). This element **MUST** be specified. This element is of type [Body](#).

Following is the parent element of the **ReportSection.Body** element.

Parent elements
ReportSection

The following is the XML Schema definition of the **ReportSection.Body** element.

```
<xsd:element name="Body" type="BodyType" minOccurs="1" />
```

### 2.6.2 ReportSection.Page

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **ReportSection.Page** element specifies the page layout information of a [ReportSection](#). This element **MUST** be specified. This element is of type [Page](#).

Following is the parent element of the **ReportSection.Page** element.

Parent elements
ReportSection

The following is the XML Schema definition of the **ReportSection.Page** element.

```
<xsd:element name="Page" type="PageType" minOccurs="1" />
```

### 2.6.3 ReportSection.Width

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **ReportSection.Width** element specifies the width of a [ReportSection](#). The **ReportSection.Width** element MUST be specified, and its value MUST be an [RdlSize](#).

Following is the parent element of the **ReportSection.Width** element.

Parent elements
ReportSection

The following is the XML Schema definition of the **ReportSection.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="1" />
```

### 2.6.4 ReportSection.DataElementName

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **ReportSection.DataElementName** element specifies the name to use for the element that represents this [ReportSection](#) in a **data rendering**. The **ReportSection.DataElementName** element is optional. If this element is present, its value MUST be a **CLS-compliant identifier** [\[UTR15\]](#).

Following is the parent element of the **ReportSection.DataElementName** element.

Parent elements
ReportSection

The following is the XML Schema definition of the **ReportSection.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.6.5 ReportSection.DataElementOutput

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **ReportSection.DataElementOutput** element specifies whether the data of a [ReportSection](#) appears in a **data rendering**. The **ReportSection.DataElementOutput** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). The value of this element is one of the following:

**Auto:** Specifies the default setting for how a **report section** will appear in a data rendering, which is the same as "NoOutput". This is the default value.

**Output:** Specifies that the report section appears in a data rendering output.

**NoOutput:** Specifies that the report section does not appear in a data rendering output.

**ContentsOnly:** Specifies that the report section itself does not appear in a data rendering output, but that its inner contents appear in a data rendering output.

If this element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ReportSection.DataElementOutput** element.

Parent elements
ReportSection

The following is the XML Schema definition of the **ReportSection.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.6.6 ReportSection.Name

*Applies to [RDL 2011/01](#)*

The **ReportSection.Name** attribute specifies the name to use for a [ReportSection](#). The **ReportSection.Name** attribute MUST be specified and MUST NOT be specified more than once. The value of this attribute MUST be a **CLS-compliant identifier** [\[UTR15\]](#).

Following is the parent element of the **ReportSection.Name** attribute.

Parent elements
ReportSection

The following is the XML Schema definition of the **ReportSection.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:string" />
```

## 2.6.7 ReportSection.LayoutDirection

*Applies to [RDL 2012/01](#)*

The **ReportSection.LayoutDirection** element specifies the layout direction to use for a [ReportSection](#). The **ReportSection.LayoutDirection** attribute is optional. If the **ReportSection.LayoutDirection** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**LTR:** Specifies that the layout for the **ReportSection** is left to right. This is the default value.

**RTL:** Specifies that the layout for the **ReportSection** is right to left.

If the **ReportSection** element is not present, its value is interpreted as "LTR".

**Note** The "RTL" layout-direction value can be specified only when using the RDL 2010/01 macro-versioned schema with the addition of the RDL 2012/01 micro-versioned schema.

Following is the parent element of the **ReportSection.LayoutDirection** element.

Parent elements
ReportSection

The following is the XML Schema definition of the **ReportSection.LayoutDirection** element.

```
<xsd:element name="LayoutDirection">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="LTR"/>
      <xsd:enumeration value="RTL"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.7 Body

The **Body** element defines the structure and layout information for the **Body** of a **report** in [RDL 2003/10](#), [RDL 2005/01](#), and [RDL 2008/01](#), or for a [ReportSection](#) in [RDL 2010/01](#) and [RDL 2016/01](#).

The following are the parent elements and child elements of the **Body** element.

Parent elements
<a href="#">Report</a>
ReportSection

Child elements
<a href="#">Body.Columns</a>
<a href="#">Body.ColumnSpacing</a>
<a href="#">Body.Style</a>
<a href="#">Body.Height</a>
<a href="#">Body.ReportItems</a>

The following is the XML Schema definition of the **Body** element in RDL 2003/10 and RDL 2005/01.

```
<xsd:complexType name="BodyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="Columns" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="ColumnSpacing" type="SizeType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **Body** element in RDL 2008/01.



```

<xsd:complexType name="BodyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="1" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **Body** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="BodyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="1" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

## 2.7.1 Body.Columns

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Body.Columns** element specifies the default number of columns used for a multi-column rendering of a [Report](#). This element is optional. If this element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2](#) section 3.3.17) that is greater than or equal to 1. [<4>](#) If this element is not present, its value is interpreted as 1.

Following is the parent element of the **Body.Columns** element.

Parent elements
<a href="#">Body</a>

The following is the XML Schema definition of the **Body.Columns** element.

```

<xsd:element name="Columns" type="xsd:unsignedInt" minOccurs="0" />

```

## 2.7.2 Body.ColumnSpacing

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Body.ColumnSpacing** element specifies the spacing between each column in a multi-column rendering of a [Report](#). The **Body.ColumnSpacing** element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as 0.5in.

Following is the parent element of the **Body.ColumnSpacing** element.

Parent elements
<a href="#">Body</a>

The following is the XML Schema definition of the **Body.ColumnSpacing** element.

```
<xsd:element name="ColumnSpacing" type="SizeType" minOccurs="0" />
```

### 2.7.3 Body.Style

The **Body.Style** element specifies style information for the [Body](#) of a report in [RDL 2003/10](#), [RDL 2005/01](#), and [RDL 2008/01](#), or for a [ReportSection](#) in [RDL 2010/01](#) and [RDL 2016/01](#). The **Body.Style** element is optional. This element is of type [Style](#).

Following is the parent element of the **Body.Style** element.

Parent elements
Body

The following is the XML Schema definition of the **Body.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.7.4 Body.Height

The **Body.Height** element specifies the height of the [Body](#) of a report in [RDL 2003/10](#), [RDL 2005/01](#), and [RDL 2008/01](#), or of a **report section** in [RDL 2010/01](#) and [RDL 2016/01](#). This element **MUST** be specified, and its value **MUST** be an [RdlSize](#).

Following is the parent element of the **Body.Height** element.

Parent elements
Body

The following is the XML Schema definition of the **Body.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="1" />
```

### 2.7.5 Body.ReportItems

The **Body.ReportItems** element specifies the top-level [Report](#) contained in the [Body](#) of a report in [RDL 2003/10](#), [RDL 2005/01](#), and [RDL 2008/01](#), or in a [ReportSection](#) in [RDL 2010/01](#) and [RDL 2016/01](#). The **Body.ReportItems** element is optional. This element is of type [ReportItems](#).

Following is the parent element of the **Body.ReportItems** element.

Parent elements
Body

The following is the XML Schema definition of the **Body.ReportItems** element.

```
<xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
```

## 2.8 Page

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Page** element specifies the structure and layout information for the pages in a report in RDL 2008/01, or for a [ReportSection](#) in RDL 2010/01 and RDL 2016/01. This element MUST be specified.

The following are the parent elements and the child elements of the **Page** element.

Parent elements
<a href="#">Report</a>
ReportSection

Child elements
<a href="#">Page.Columns</a>
<a href="#">Page.ColumnSpacing</a>
<a href="#">Page.BottomMargin</a>
<a href="#">Page.InteractiveHeight</a>
<a href="#">Page.InteractiveWidth</a>
<a href="#">Page.LeftMargin</a>
<a href="#">Page.PageFooter</a>
<a href="#">Page.PageHeader</a>
<a href="#">Page.PageHeight</a>
<a href="#">Page.PageWidth</a>
<a href="#">Page.RightMargin</a>
<a href="#">Page.Style</a>
<a href="#">Page.TopMargin</a>

The following is the XML Schema definition of the **Page** element in RDL 2008/01.

```
<xsd:complexType name="PageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="PageHeader" type="PageSectionType" minOccurs="0" />
    <xsd:element name="PageFooter" type="PageSectionType" minOccurs="0" />
    <xsd:element name="PageHeight" type="SizeType" minOccurs="0" />
    <xsd:element name="PageWidth" type="SizeType" minOccurs="0" />
    <xsd:element name="InteractiveHeight" type="SizeType" minOccurs="0" />
    <xsd:element name="InteractiveWidth" type="SizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="RightMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="TopMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="BottomMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="Columns" type="xsd:int" minOccurs="0" />
    <xsd:element name="ColumnSpacing" type="SizeType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

```

    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **Page** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="PageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="PageHeader" type="PageSectionType" minOccurs="0" />
    <xsd:element name="PageFooter" type="PageSectionType" minOccurs="0" />
    <xsd:element name="PageHeight" type="SizeType" minOccurs="0" />
    <xsd:element name="PageWidth" type="SizeType" minOccurs="0" />
    <xsd:element name="InteractiveHeight" type="SizeType" minOccurs="0" />
    <xsd:element name="InteractiveWidth" type="SizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="RightMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="TopMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="BottomMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="Columns" type="xsd:int" minOccurs="0" />
    <xsd:element name="ColumnSpacing" type="SizeType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

## 2.8.1 Page.Columns

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Page.Columns** element specifies the default number of columns used to render a [Report](#) in RDL 2008/01 or a [ReportSection](#) in RDL 2010/01 and RDL 2016/01. The **Page.Columns** element is optional. If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) that is greater than or equal to 1. [<5>](#) If this element is not present, its value is interpreted as 1.

Following is the parent element of the **Page.Columns** element.

Parent elements
<a href="#">Page</a>

The following is the XML Schema definition of the **Page.Columns** element.

```

<xsd:element name="Columns" type="xsd:int" minOccurs="0" />

```

## 2.8.2 Page.ColumnSpacing

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Page.ColumnSpacing** element specifies the spacing between each column for a multi-column rendering of a [Report](#) in RDL 2008/01 or of a [ReportSection](#) in RDL 2010/01 and RDL 2016/01. The **Page.ColumnSpacing** element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as 0.5in.

Following is the parent element of the **Page.ColumnSpacing** element.

Parent elements
<a href="#">Page</a>

The following is the XML Schema definition of the **Page.ColumnSpacing** element.

```
<xsd:element name="ColumnSpacing" type="SizeType" minOccurs="0" />
```

### 2.8.3 Page.BottomMargin

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Page.BottomMargin** element specifies the width of the bottom margin of a [Page](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as 0in.

Following is the parent element of the **Page.BottomMargin** element.

Parent elements
<a href="#">Page</a>

The following is the XML Schema definition of the **Page.BottomMargin** element.

```
<xsd:element name="BottomMargin" type="SizeType" minOccurs="0" />
```

### 2.8.4 Page.InteractiveHeight

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Page.InteractiveHeight** element specifies the default height for rendering a [Report](#) by an **interactive renderer**. The **Page.InteractiveHeight** element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted to be the value of [Page.PageHeight](#).

Following is the parent element of the **Page.InteractiveHeight** element.

Parent elements
<a href="#">Page</a>

The following is the XML Schema definition of the **Page.InteractiveHeight** element.

```
<xsd:element name="InteractiveHeight" type="SizeType" minOccurs="0" />
```

### 2.8.5 Page.InteractiveWidth

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Page.InteractiveWidth** element specifies the default width for rendering a [Report](#) by an **interactive renderer**. The **Page.InteractiveWidth** element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted to be the value of [Page.PageWidth](#).

Following is the parent element of the **Page.InteractiveWidth** element.

Parent elements
<a href="#">Page</a>

The following is the XML Schema definition of the **Page.InteractiveWidth** element.

```
<xsd:element name="InteractiveWidth" type="SizeType" minOccurs="0" />
```

## 2.8.6 Page.LeftMargin

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Page.LeftMargin** element specifies the width of the left margin of a [Page](#). The **Page.LeftMargin** element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as 0in.

Following is the parent element of the **Page.LeftMargin** element.

Parent elements
<a href="#">Page</a>

The following is the XML Schema definition of the **Page.LeftMargin** element.

```
<xsd:element name="LeftMargin" type="SizeType" minOccurs="0" />
```

## 2.8.7 Page.PageFooter

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Page.PageFooter** element specifies a footer that is rendered at the bottom of each rendered [Page](#) in RDL 2008/01 or at the bottom of [ReportSection](#) in RDL 2010/01 and RDL 2016/01. The **Page.PageFooter** element is optional. This element is of type [PageSection](#).

Following is the parent element of the **Page.PageFooter** element.

Parent elements
<a href="#">Page</a>

The following is the XML Schema definition of the **Page.PageFooter** element.

```
<xsd:element name="PageFooter" type="PageSectionType" minOccurs="0" />
```

## 2.8.8 Page.PageHeader

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Page.PageHeader** element specifies a header that is rendered at the top of each rendered [Page](#) in a report in RDL 2008/01, or at the top of the [ReportSection](#) in RDL 2010/01 and RDL 2016/01. The **Page.PageHeader** element is optional. This element is of type [PageSection](#).

Following is the parent element of the **Page.PageHeader** element.

Parent elements
Page

The following is the XML Schema definition of the **Page.PageHeader** element.

```
<xsd:element name="PageHeader" type="PageSectionType" minOccurs="0" />
```

## 2.8.9 Page.PageHeight

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Page.PageHeight** element specifies the default height for rendering, in a **physical-page oriented renderer**, each [Page](#) at the top of a report in RDL 2008/01, or each [ReportSection](#) in RDL 2010/01 and RDL 2016/01. The **Page.PageHeight** element is optional. If this element is present, its value **MUST** be an [RdlSize](#). If this element is not present, its value is interpreted as 11in.

Following is the parent element of the **Page.PageHeight** element.

Parent elements
Page

The following is the XML Schema definition of the **Page.PageHeight** element.

```
<xsd:element name="PageHeight" type="SizeType" minOccurs="0" />
```

## 2.8.10 Page.PageWidth

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Page.PageWidth** element specifies the default width for rendering, in a **physical-page oriented renderer**, each [Page](#) in a [Report](#) in RDL 2008/01, or the **report section** in RDL 2010/01 and RDL 2016/01. The **Page.PageWidth** element is optional. If this element is present, its value **MUST** be an [RdlSize](#). If this element is not present, its value is interpreted as 8.5in.

Following is the parent element of the **Page.PageWidth** element.

Parent elements
Page

The following is the XML Schema definition of the **Page.PageWidth** element.

```
<xsd:element name="PageWidth" type="SizeType" minOccurs="0" />
```

## 2.8.11 Page.RightMargin

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Page.RightMargin** element specifies the width of the right margin of a [Page](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as 0in.

Following is the parent element of the **Page.RightMargin** element.

Parent elements
Page

The following is the XML Schema definition of the **Page.RightMargin** element.

```
<xsd:element name="RightMargin" type="SizeType" minOccurs="0" />
```

## 2.8.12 Page.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Page.Style** element specifies style information for a [Page](#) in a report in RDL 2008/01, or for a [ReportSection](#) in RDL 2010/01 and RDL 2016/01. The **Page.Style** element is optional. This element is of type [Style](#).

Following is the parent element of the **Page.Style** element.

Parent elements
Page

The following is the XML Schema definition of the **Page.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.8.13 Page.TopMargin

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Page.TopMargin** element specifies the height of the top margin of a [Page](#). The **Page.TopMargin** element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as 0in.

Following is the parent element of the **Page.TopMargin** element.

Parent elements
Page

The following is the XML Schema definition of the **Page.TopMargin** element.

```
<xsd:element name="TopMargin" type="SizeType" minOccurs="0" />
```

## 2.9 PageSection

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*



The **PageSection** element specifies the structure and layout information for the **page header** or **page footer** of a rendered [Page](#) in a report in RDL 2008/01 or a [ReportSection](#) in RDL 2010/01 and RDL 2016/01. This element is optional.

The following are elements that are referenced by the **PageSection** element, its parent element, and its child elements.

Referenced by
<a href="#">Page.PageHeader</a>
<a href="#">Page.PageFooter</a>

Parent elements
<a href="#">Page</a>

Child elements
<a href="#">PageSection.Style</a>
<a href="#">PageSection.Height</a>
<a href="#">PageSection.PrintOnFirstPage</a>
<a href="#">PageSection.PrintOnLastPage</a>
<a href="#">PageSection.PrintBetweenSections</a>
<a href="#">PageSection.ReportItems</a>

The following is the XML Schema definition of the **PageSection** element in RDL 2008/01.

```
<xsd:complexType name="PageSectionType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="PrintOnFirstPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PrintOnLastPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **PageSection** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="PageSectionType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="PrintOnFirstPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PrintOnLastPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PrintBetweenSections" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
</xsd:complexType>
```

```
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.9.1 PageSection.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PageSection.Style** element specifies the style information for a [PageSection](#). This element is optional. This element is of type [Style](#).

Following is the parent element of the **PageSection.Style** element.

Parent elements
PageSection

The following is the XML Schema definition of the **PageSection.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.9.2 PageSection.Height

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PageSection.Height** element specifies the height of a [PageSection](#). This element MUST be specified, and its value MUST be an [RdlSize](#).

Following is the parent element of the **PageSection.Height** element.

Parent elements
PageSection

The following is the XML Schema definition of the **PageSection.Height** element.

```
<xsd:element name="Height" type="SizeType" />
```

### 2.9.3 PageSection.PrintOnFirstPage

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PageSection.PrintOnFirstPage** element specifies whether the [PageSection](#) is shown on the first rendered [Page](#) in a report in RDL 2008/01 or the [ReportSection](#) in RDL 2010/01 and RDL 2016/01. The **PageSection.PrintOnFirstPage** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **PageSection.PrintOnFirstPage** element.

Parent elements
PageSection

The following is the XML Schema definition of the **PageSection.PrintOnFirstPage** element.

```
<xsd:element name="PrintOnFirstPage" type="xsd:boolean" minOccurs="0" />
```

#### 2.9.4 PageSection.PrintOnLastPage

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PageSection.PrintOnLastPage** element specifies whether the [PageSection](#) is shown on the last rendered [Page](#) in a report in RDL 2008/01 or [ReportSection](#) in RDL 2010/01 and RDL 2016/01. The **PageSection.PrintOnLastPage** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **PageSection.PrintOnLastPage** element.

Parent elements
PageSection

The following is the XML Schema definition of the **PageSection.PrintOnLastPage** element.

```
<xsd:element name="PrintOnLastPage" type="xsd:boolean" minOccurs="0" />
```

#### 2.9.5 PageSection.PrintBetweenSections

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PageSection.PrintBetweenSections** element specifies whether the [PageSection](#) should be shown on the rendered [Page](#) in a report in RDL 2008/01 or a [ReportSection](#) in RDL 2010/01 and RDL 2016/01. The **PageSection.PrintBetweenSections** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **PageSection.PrintBetweenSections** element.

Parent elements
PageSection

The following is the XML Schema definition of the **PageSection.PrintBetweenSections** element.

```
<xsd:element name="PrintBetweenSections" type="xsd:boolean" minOccurs="0" />
```

#### 2.9.6 PageSection.ReportItems

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PageSection.ReportItems** element specifies the region that contains the elements of the page section layout. This element MUST NOT contain **report items** of type [Tablix](#), [Chart](#), [GaugePanel](#), [Map](#), or [Subreport](#). All [PageBreak](#) instances within the **PageSection.ReportItems** element are ignored.

The **PageSection.ReportItems** element is optional. This element is of type [ReportItems](#).

Following is the parent element of the **PageSection.ReportItems** element.

Parent elements
<a href="#">PageSection</a>

The following is the XML Schema definition of the **PageSection.ReportItems** element.

```
<xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
```

## 2.10 PageHeaderFooter

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **PageHeaderFooter** element specifies the structure and layout information for a **page header** or **page footer** in a [Report](#).

The following are the elements that are referenced by the **PageHeaderFooter** element, as well as its parent element and child elements.

Referenced by
<a href="#">Report.PageHeader</a>
<a href="#">Report.PageFooter</a>

Parent elements
Report

Child elements
<a href="#">PageHeaderFooter.Height</a>
<a href="#">PageHeaderFooter.PrintOnFirstPage</a>
<a href="#">PageHeaderFooter.PrintOnLastPage</a>
<a href="#">PageHeaderFooter.ReportItems</a>
<a href="#">PageHeaderFooter.Style</a>

The following is the XML Schema definition of the **PageHeaderFooter** element.

```
<xsd:complexType name="PageHeaderFooterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="PrintOnFirstPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PrintOnLastPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

</xsd:complexType>

### 2.10.1 PageHeaderFooter.Height

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **PageHeaderFooter.Height** element specifies the height of a **page header** or **page footer**. This element MUST be specified, and its value MUST be an [RdlSize](#).

Following is the parent element of the **PageHeaderFooter.Height** element.

Parent elements
<a href="#">PageHeaderFooter</a>

The following is the XML Schema definition of the **PageHeaderFooter.Height** element.

```
<xsd:element name="Height" type="SizeType" />
```

### 2.10.2 PageHeaderFooter.PrintOnFirstPage

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **PageHeaderFooter.PrintOnFirstPage** element specifies whether the **page header** or **page footer** is shown on the first rendered [Page](#) in a report. The **PageHeaderFooter.PrintOnFirstPage** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **PageHeaderFooter.PrintOnFirstPage** element.

Parent elements
<a href="#">PageHeaderFooter</a>

The following is the XML Schema definition of the **PageHeaderFooter.PrintOnFirstPage** element.

```
<xsd:element name="PrintOnFirstPage" type="xsd:boolean" minOccurs="0" />
```

### 2.10.3 PageHeaderFooter.PrintOnLastPage

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **PageHeaderFooter.PrintOnLastPage** element specifies whether the **page header** or **page footer** is shown on the last rendered [Page](#) in a report. The **PageHeaderFooter.PrintOnLastPage** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **PageHeaderFooter.PrintOnLastPage** element.

Parent elements
<a href="#">PageHeaderFooter</a>

The following is the XML Schema definition of the **PageHeaderFooter.PrintOnLastPage** element.

```
<xsd:element name="PrintOnLastPage" type="xsd:boolean" minOccurs="0" />
```

## 2.10.4 PageHeaderFooter.ReportItems

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **PageHeaderFooter.ReportItems** element specifies the region that contains the elements of the page section layout. This element is optional. This element MUST NOT contain **report items** of type [List](#), [Table](#), [Matrix](#), [Chart](#), or [Subreport](#). The **PageHeaderFooter.ReportItems** element is of type [ReportItems](#).

Following is the parent element of the **PageHeaderFooter.ReportItems** element.

Parent elements
<a href="#">PageHeaderFooter</a>

The following is the XML Schema definition of the **PageHeaderFooter.ReportItems** element.

```
<xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
```

## 2.10.5 PageHeaderFooter.Style

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **PageHeaderFooter.Style** element specifies the style information for a [PageHeaderFooter](#). The **PageHeaderFooter.Style** element is optional. This element is of type [Style](#).

Following is the parent element of the **PageHeaderFooter.Style** element.

Parent elements
<a href="#">PageHeaderFooter</a>

The following is the XML Schema definition of the **PageHeaderFooter.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.11 ReportItems

The **ReportItems** element specifies the collection of **report items** used to define the contents of a region of a [Report](#).

The following are the parent and child elements of the **ReportItems** element.

Parent elements
<a href="#">Body</a>
<a href="#">PageSection</a>
<a href="#">PageHeaderFooter</a>
<a href="#">Rectangle</a>

Parent elements
<a href="#">List</a>
<a href="#">MatrixCell</a>
<a href="#">Corner</a>
<a href="#">DynamicColumns</a>
<a href="#">DynamicRows</a>
<a href="#">StaticColumn</a>
<a href="#">StaticRow</a>
<a href="#">Subtotal</a>
<a href="#">TableCell</a>
<a href="#">CustomReportItem</a>

Child elements
<a href="#">ReportItems.Chart</a>
<a href="#">ReportItems.CustomReportItem</a>
<a href="#">ReportItems.GaugePanel</a>
<a href="#">ReportItems.Image</a>
<a href="#">ReportItems.Line</a>
<a href="#">ReportItems.Map</a>
<a href="#">ReportItems.Rectangle</a>
<a href="#">ReportItems.Subreport</a>
<a href="#">ReportItems.Tablix</a>
<a href="#">ReportItems.Textbox</a>
<a href="#">ReportItems.List</a>
<a href="#">ReportItems.Matrix</a>
<a href="#">ReportItems.Table</a>

The following is the XML Schema definition of the **ReportItems** element in [RDL 2003/10](#) and [RDL 2005/01](#).

```
<xsd:complexType name="ReportItemsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Line" type="LineType" />
    <xsd:element name="Rectangle" type="RectangleType" />
    <xsd:element name="Textbox" type="TextboxType" />
    <xsd:element name="Image" type="ImageType" />
    <xsd:element name="Subreport" type="SubreportType" />
    <xsd:element name="Chart" type="ChartType" />
    <xsd:element name="List" type="ListType" />
  </xsd:choice>
</xsd:complexType>
```

```

<xsd:element name="Table" type="TableType" />
<xsd:element name="Matrix" type="MatrixType" />
<xsd:element name="CustomReportItem" type="CustomReportItemType" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **ReportItems** element in [RDL 2008/01](#).

```

<xsd:complexType name="ReportItemsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Line" type="LineType" />
    <xsd:element name="Rectangle" type="RectangleType" />
    <xsd:element name="Textbox" type="TextboxType" />
    <xsd:element name="Image" type="ImageType" />
    <xsd:element name="Subreport" type="SubreportType" />
    <xsd:element name="Chart" type="ChartType" />
    <xsd:element name="GaugePanel" type="GaugePanelType" />
    <xsd:element name="Tablix" type="TablixType" />
    <xsd:element name="CustomReportItem" type="CustomReportItemType" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **ReportItems** element in [RDL 2010/01](#) and [RDL 2016/01](#).

```

<xsd:complexType name="ReportItemsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Line" type="LineType" />
    <xsd:element name="Rectangle" type="RectangleType" />
    <xsd:element name="Textbox" type="TextboxType" />
    <xsd:element name="Image" type="ImageType" />
    <xsd:element name="Subreport" type="SubreportType" />
    <xsd:element name="Chart" type="ChartType" />
    <xsd:element name="GaugePanel" type="GaugePanelType" />
    <xsd:element name="Map" type="MapType" />
    <xsd:element name="Tablix" type="TablixType" />
    <xsd:element name="CustomReportItem" type="CustomReportItemType" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.11.1 ReportItems.Chart

The **ReportItems.Chart** element specifies a **chart** as a **member** of a [ReportItems](#) collection. This element is optional and can be specified more than once. The **ReportItems.Chart** element is of type [Chart](#).

Following is the parent element of the **ReportItems.Chart** element.

Parent elements
ReportItems

The following is the XML Schema definition of the **ReportItems.Chart** element.



```
<xsd:element name="Chart" type="ChartType" />
```

### 2.11.2 ReportItems.CustomReportItem

The **ReportItems.CustomReportItem** element specifies a **custom report item** as a member of a [ReportItems](#) collection. This element is optional and can be specified more than once. The **ReportItems.CustomReportItem** element is of type [CustomReportItem](#).

Following is the parent element of the **ReportItems.CustomReportItem** element.

Parent elements
ReportItems

The following is the XML Schema definition of the **ReportItems.CustomReportItem** element.

```
<xsd:element name="CustomReportItem" type="CustomReportItemType" />
```

### 2.11.3 ReportItems.GaugePanel

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ReportItems.GaugePanel** element specifies a **gauge panel** as a member of a [ReportItems](#) collection. This element is optional and can be specified more than once. The **ReportItems.GaugePanel** element is of type [GaugePanel](#).

Parent elements
ReportItems

The following is the XML Schema definition of the **ReportItems.GaugePanel** element.

```
<xsd:element name="GaugePanel" type="GaugePanelType" />
```

### 2.11.4 ReportItems.Image

The **ReportItems.Image** element specifies an **image** as a member of a [ReportItems](#) collection. This element is optional and can be specified more than once. The **ReportItems.Image** element is of type [Image](#).

Following is the parent element of the **ReportItems.Image** element.

Parent elements
ReportItems

The following is the XML Schema definition of the **ReportItems.Image** element.

```
<xsd:element name="Image" type="ImageType" />
```

### 2.11.5 ReportItems.Line

The **ReportItems.Line** element specifies a line as a member of a [ReportItems](#) collection. This element is optional and can be specified more than once. The **ReportItems.Line** element is of type [Line](#).

Following is the parent element of the **ReportItems.Line** element.

Parent elements
ReportItems

The following is the XML Schema definition of the **ReportItems.Line** element.

```
<xsd:element name="Line" type="LineType" />
```

### 2.11.6 ReportItems.Map

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **ReportItems.Map** element specifies a **map** as a member of a [ReportItems](#) collection. This element is optional and can be specified more than once. The **ReportItems.Map** element is of type [Map](#).

Following is the parent element of the **ReportItems.Map** element.

Parent elements
ReportItems

The following is the XML Schema definition of the **ReportItems.Map** element.

```
<xsd:element name="Map" type="MapType" />
```

### 2.11.7 ReportItems.Rectangle

The **ReportItems.Rectangle** element specifies a **rectangle** as a member of a [ReportItems](#) collection. The **ReportItems.Rectangle** element is optional and can be specified more than once. If this element is present, it is of type [Rectangle](#).

Following is the parent element of the **ReportItems.Rectangle** element.

Parent elements
ReportItems

The following is the XML Schema definition of the **ReportItems.Rectangle** element.

```
<xsd:element name="Rectangle" type="RectangleType" />
```

### 2.11.8 ReportItems.Subreport

The **ReportItems.Subreport** element specifies a subreport as a member of a [ReportItems](#) collection. This element is optional and can be specified more than once. The **ReportItems.Subreport** element is of type [Subreport](#).

Following is the parent element of the **ReportItems.Subreport** element.

Parent elements
ReportItems

The following is the XML Schema definition of the **ReportItems.Subreport** element.

```
<xsd:element name="Subreport" type="SubreportType" />
```

### 2.11.9 ReportItems.Tablix

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ReportItems.Tablix** element specifies a **tablix** as a member of a [ReportItems](#) collection. This element is optional and can be specified more than once. The **ReportItems.Tablix** element is of type [Tablix](#).

Following is the parent element of the **ReportItems.Tablix** element.

Parent elements
ReportItems

The following is the XML Schema definition of the **ReportItems.Tablix** element.

```
<xsd:element name="Tablix" type="TablixType" />
```

### 2.11.10 ReportItems.Textbox

The **ReportItems.Textbox** element specifies a **text box** as a member of a [ReportItems](#) collection. This element is optional and can be specified more than once. The **ReportItems.Textbox** element is of type [Textbox](#).

Following is the parent element of the **ReportItems.Textbox** element.

Parent elements
ReportItems

The following is the XML Schema definition of the **ReportItems.Textbox** element.

```
<xsd:element name="Textbox" type="TextboxType" />
```

### 2.11.11 ReportItems.List

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **ReportItems.List** element specifies a **list** as a member of a [ReportItems](#) collection. This element is optional and can be specified more than once. The **ReportItems.List** element is of type [List](#).

Following is the parent element of the **ReportItems.List** element.

Parent elements
ReportItems

The following is the XML Schema definition of the **ReportItems.List** element.

```
<xsd:element name="List" type="ListType" />
```

### 2.11.12 ReportItems.Matrix

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ReportItems.Matrix** element specifies a **matrix** as a member of a [ReportItems](#) collection. This element is optional and can be specified more than once. The **ReportItems.Matrix** element is of type [Matrix](#).

Following is the parent element of the **ReportItems.Matrix** element.

Parent elements
ReportItems

The following is the XML Schema definition of the **ReportItems.Matrix** element.

```
<xsd:element name="Matrix" type="MatrixType" />
```

### 2.11.13 ReportItems.Table

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ReportItems.Table** element specifies a **table** as a member of a [ReportItems](#) collection. This element is optional and can be specified more than once. The **ReportItems.Table** element is of type [Table](#).

Following is the parent element of the **ReportItems.Table** element.

Parent elements
ReportItems

The following is the XML Schema definition of the **ReportItems.Table** element.

```
<xsd:element name="Table" type="TableType" />
```

## 2.12 Image

The **Image** element specifies properties for an **image** in a [Report](#).

The following are the parent elements, attributes, and child elements of the **Image** element.

Parent elements
<a href="#">ReportItems</a>
<a href="#">CellContents</a>
<a href="#">CustomReportItem.AltReportItem</a>

Attributes
<a href="#">Image.Name</a>

Child elements
<a href="#">Image.Style</a>
<a href="#">Image.ActionInfo</a>
<a href="#">Image.Bookmark</a>
<a href="#">Image.CustomProperties</a>
<a href="#">Image.DataElementName</a>
<a href="#">Image.DataElementOutput</a>
<a href="#">Image.DocumentMapLabel</a>
<a href="#">Image.Height</a>
<a href="#">Image.Left</a>
<a href="#">Image.RepeatWith</a>
<a href="#">Image.ToolTip</a>
<a href="#">Image.Top</a>
<a href="#">Image.Visibility</a>
<a href="#">Image.Width</a>
<a href="#">Image.ZIndex</a>
<a href="#">Image.MIMETYPE</a>
<a href="#">Image.Sizing</a>
<a href="#">Image.Source</a>
<a href="#">Image.Value</a>

**Applies to** [RDL 2011/01](#)

Child elements
<a href="#">Image.Tag</a>

**Applies to [RDL 2012/01](#)**

<b>Child elements</b>
<a href="#">Image.EmbeddingMode</a>

**Applies to [RDL 2013/01](#)**

<b>Child elements</b>
<a href="#">Image.Tags</a>

The following is the XML Schema definition of the **Image** element.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, 5.6, and 5.7, respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:complexType name="ImageType" >
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="Source">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="External" />
          <xsd:enumeration value="Embedded" />
          <xsd:enumeration value="Database" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="Sizing" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="AutoSize" />
          <xsd:enumeration value="Fit" />
          <xsd:enumeration value="FitProportional" />
          <xsd:enumeration value="Clip" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>
```

```

    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.12.1 Image.Name

The **Image.Name** attribute specifies a unique identifier for an [Image](#) instance. This attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique for all **report items** in a [Report](#).

Following is the parent element of the **Image.Name** attribute.

Parent elements
Image

The following is the XML Schema definition of the **Image.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.12.2 Image.Style

The **Image.Style** element specifies style properties for an [Image](#). This element is optional. This element is of type [Style](#).

Following is the parent element of the **Image.Style** element.

Parent elements
Image

The following is the XML Schema definition of the **Image.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.12.3 Image.ActionInfo

The **Image.ActionInfo** element specifies **actions** (such as hyperlinks) that are associated with an [Image](#). This element is optional. This element is of type [ActionInfo](#).

Following is the parent element of the **Image.ActionInfo** element.

Parent elements
Image

The following is the XML Schema definition of the **Image.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

In [RDL 2003/10](#) and [RDL 2005/01](#), the equivalent element of **Image.ActionInfo** is **Image.Action**, which is of type [Action](#).

#### 2.12.4 Image.Bookmark

The **Image.Bookmark** element specifies a **bookmark** for an [Image](#) that can be linked to via a bookmark **action**. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Image.Bookmark** element.

Parent elements
<a href="#">Image</a>

The following is the XML Schema definition of the **Image.Bookmark** element.

```
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
```

#### 2.12.5 Image.CustomProperties

The **Image.CustomProperties** element specifies custom information for an [Image](#) instance that will be handed to a report rendering component. This element is optional. This element is of type [CustomProperties](#).

Following is the parent element of the **Image.CustomProperties** element.

Parent elements
<a href="#">Image</a>

The following is the XML Schema definition of the **Image.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

In [RDL 2003/10](#), the equivalent element of **Image.CustomProperties** is **Image.Custom**, which is of type [Custom](#).

#### 2.12.6 Image.DataElementName

The **Image.DataElementName** element is ignored.

Following is the parent element of the **Image.DataElementName** element.

Parent elements
<a href="#">Image</a>

The following is the XML Schema definition of the **Image.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```



### 2.12.7 Image.DataElementOutput

The **Image.DataElementOutput** element is ignored.

Following is the parent element of the **Image.DataElementOutput** element.

Parent elements
<a href="#">Image</a>

The following is the XML Schema definition of the **Image.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0"
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.12.8 Image.DocumentMapLabel

The **Image.DocumentMapLabel** element specifies a label that is used to identify an [Image](#) instance within the client UI in order to provide a user-friendly label for searching. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. This element MUST NOT be used within an **Image** that is contained within a [PageSection](#) or [PageHeaderFooter](#).

Following is the parent element of the **Image.DocumentMapLabel** element.

Parent elements
<a href="#">Image</a>

The following is the XML Schema definition of the **Image.DocumentMapLabel** element.

```
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
```

In [RDL 2003/10](#) and [RDL 2005/01](#), the equivalent element of **Image.DocumentMapLabel** is **Image.Label**. The following is the XML Schema definition of the **Image.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

### 2.12.9 Image.Height

The **Image.Height** element specifies the height of an [Image](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as the height of the image's container, such as a **Rectangle** or **Body**, minus the value of the [Image.Top](#) element, if specified.

Following is the parent element of the **Image.Height** element.

Parent elements
Image

The following is the XML Schema definition of the **Image.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="0" />
```

### 2.12.10 Image.Left

The **Image.Left** element specifies the distance of an [Image](#) from the left of the **image** container, such as a **Rectangle** or **Body**. This element is optional. If this element is present, its value MUST be an [RdlSize](#).

Following is the parent element of the **Image.Left** element.

Parent elements
Image

The following is the XML Schema definition of the **Image.Left** element.

```
<xsd:element name="Left" type="SizeType" minOccurs="0" />
```

### 2.12.11 Image.RepeatWith

The **Image.RepeatWith** element specifies the name of a **data region** in which the [Image](#) is repeated if the data region spans multiple pages. The data region MUST be in the same [ReportItems](#) collection as the **image**. If the image is in a [PageSection](#) or a [PageHeaderFooter](#), the **Image.RepeatWith** element is ignored.

The **Image.RepeatWith** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Following is the parent element of the **Image.RepeatWith** element.

Parent elements
Image

The following is the XML Schema definition of the **Image.RepeatWith** element.

```
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
```

### 2.12.12 Image.ToolTip

The **Image.ToolTip** element specifies the tooltip text for an [Image](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **Image.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Image.ToolTip** element.

Parent elements
Image

The following is the XML Schema definition of the **Image.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

### 2.12.13 Image.Top

The **Image.Top** element specifies the distance of an [Image](#) from the top of the **image** container, such as a **Rectangle** or **Body**. This element is optional. If this element is present, its value MUST be an [RdlSize](#).

Following is the parent element of the **Image.Top** element.

Parent elements
Image

The following is the XML Schema definition of the **Image.Top** element.

```
<xsd:element name="Top" type="SizeType" minOccurs="0" />
```

### 2.12.14 Image.Visibility

The **Image.Visibility** element specifies whether an [Image](#) is hidden. This element is optional. This element is of type [Visibility](#).

Following is the parent element of the **Image.Visibility** element.

Parent elements
Image

The following is the XML Schema definition of the **Image.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

### 2.12.15 Image.Width

The **Image.Width** element specifies the width of an [Image](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as the width of the **image's** container, such as **Rectangle** or **Body** minus the value of the [Image.Left](#) element, if specified.

Following is the parent element of the **Image.Width** element.

Parent elements
Image

The following is the XML Schema definition of the **Image.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="0" />
```

### 2.12.16 Image.ZIndex

The **Image.ZIndex** element specifies the drawing order of an [Image](#) within its container. This element is optional. If this element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17). If this element is not present, its value is interpreted as 0. The value of this element MUST be greater than or equal to 0 and less than or equal to 2147483647.

Following is the parent element of the **Image.ZIndex** element.

Parent elements
Image

The following is the XML Schema definition of the **Image.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.12.17 Image.MIMETYPE

The **Image.MIMETYPE** element specifies the image format of an [Image](#). The **Image.MIMETYPE** element is optional. If this element is present, its value MUST be a [ReportMIMETYPE](#).

If the value of the peer element [Image.Source](#) is set to a value other than "Database", the **Image.MIMETYPE** element is ignored. If the value of the peer element **Image.Source** is set to "Database", the **Image.MIMETYPE** element MUST be specified.

Following is the parent element of the **Image.MIMETYPE** element.

Parent elements
Image

The following is the XML Schema definition of the **Image.MIMETYPE** element.

```
<xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
```

### 2.12.18 Image.Sizing

The **Image.Sizing** element specifies the behavior of an [Image](#) instance if the actual image does not fit within the specified size of the image as denoted by [Image.Height](#) and [Image.Width](#). This element is optional. If the **Image.Sizing** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is one of the following:

**AutoSize:** Specifies that the border of the image instance will grow or shrink to accommodate the dimensions of the actual image.

**Fit:** Specifies that the actual image is resized to exactly match both the **Image.Height** and **Image.Width** dimensions.

**FitProportional:** Specifies that the actual image is resized to fit within the image instance dimensions, while preserving the aspect ratio.

**Clip:** Specifies that the actual image is clipped to fit the dimensions of the image instance.

If the **Image.Sizing** element is not present, its value is interpreted as "AutoSize".

Following is the parent element of the **Image.Sizing** element.

Parent elements
Image

The following is the XML Schema definition of the **Image.Sizing** element.

```
<xsd:element name="Sizing" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="AutoSize" />
      <xsd:enumeration value="Fit" />
      <xsd:enumeration value="FitProportional" />
      <xsd:enumeration value="Clip" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.12.19 Image.Source

The **Image.Source** element specifies the type of source for an [Image](#). This element MUST be specified. The value of this element MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) that is one of the following:

**External:** Specifies that the peer [Image.Value](#) element MUST contain a **String** constant or an expression that evaluates to the location of an image.

**Embedded:** Specifies that the peer **Image.Value** element MUST contain a **String** constant or an expression that evaluates to the name of an **embedded image** in the report.

**Database:** Specifies that the peer **Image.Value** element MUST contain an expression (such as a field in the database) that evaluates to the binary data for an image.

Following is the parent element of the **Image.Source** element.

Parent elements
Image

The following is the XML Schema definition of the **Image.Source** element.

```
<xsd:element name="Source">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="External" />
      <xsd:enumeration value="Embedded" />
      <xsd:enumeration value="Database" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.12.20 Image.Value

The **Image.Value** element specifies either the location or the actual data of an image, depending on the value of the peer [Image.Source](#) element.

If the peer **Image.Source** element is set to "External", the value of the **Image.Value** element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) constant or an expression that evaluates to the location of an [Image](#). This location MUST be a [ReportPath](#) or an [RdlURL](#) value.

If the peer **Image.Source** element is set to "Embedded", the value of the **Image.Value** element MUST be a **String** constant or an expression that evaluates to the name of an [EmbeddedImage](#) in the report.

If the peer **Image.Source** element is set to "Database", the value of the **Image.Value** element MUST be a **String** constant or an expression that evaluates to the binary data for an image.

If the **Image.Value** element has an empty value, an image MUST NOT be displayed.

Following is the parent element of the **Image.Value** element.

Parent elements
<a href="#">Image</a>

The following is the XML Schema definition of the **Image.Value** element.

```
<xsd:element name="Value" type="xsd:string" />
```

## 2.12.21 Image.Tag

**Applies to** [RDL 2011/01](#)

The **Image.Tag** element contains arbitrary additional information about the image that clients can use to perform custom processing. The **Image.Tag** element is optional and MUST NOT be specified more than once. The value of the **Image.Tag** element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) constant or an expression that evaluates to a **variant**.

Following is the parent element of the **Image.Tag** element.

Parent elements
<a href="#">Image</a>

The following is the XML Schema definition of the **Image.Tag** element.

```
<xsd:element name="Tag" type="xsd:string" />
```

## 2.12.22 Image.Tags

**Applies to** [RDL 2013/01](#)

The **Image.Tags** element supersedes the [Image.Tag](#) element of [RDL 2011/01](#). The **Image.Tags** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type **Tags**. The **Image.Tags** element is a collection of **Tag** elements. The value of each **Tag** element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) constant or an expression that evaluates to a

**variant.** The **Image.Tags** element contains a collection of arbitrary additional information about the image that clients can use to perform custom processing.

Following is the parent element of the **Image.Tags** element.

Parent elements
<a href="#">Image</a>

The following is the XML Schema definition of the **Image.Tags** element.

```
<xsd:element name="Tags">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Tag" type="xsd:string" minOccurs="1" maxOccurs="unbounded" />
      <xsd:any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
</xsd:element>
```

### 2.12.23 Image.EmbeddingMode

**Applies to** [RDL 2012/01](#)

The **Image.EmbeddingMode** element specifies the embedding type for an [Image](#). The **Image.EmbeddingMode** element is optional and MUST NOT be specified if [Image.Source](#) is not embedded. If **Image.EmbeddingMode** is present, the value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). The value of the **Image.EmbeddingMode** element MUST be one of the following:

**Inline:** Specifies that the **report definition** includes the image.

**Package:** Specifies that [Image.Value](#) specifies the name of an image file that is outside the report. Typically, the package includes the image file as a package. For more details, see [\[MS-DPRDL\]](#) section 2.2.1 and [\[ECMA-376-2/2\]](#).

If the **Image.EmbeddingMode** element is not present, its value is interpreted as "Inline".

Following is the parent element of the **Image.EmbeddingMode** element.

Parent elements
<a href="#">Image</a>

The following is the XML Schema definition of the **Image.EmbeddingMode** element.

```
<xsd:element name="EmbeddingMode">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Inline"/>
      <xsd:enumeration value="Package"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.13 Line

The **Line** element specifies properties to draw a **line** in a report.

One endpoint of the line, the origin, is specified by the [Line.Left](#) and [Line.Top](#) elements. The other endpoint of the line is specified, relative to the origin, by the [Line.Height](#) and [Line.Width](#) elements.

Negative heights and widths allow for lines that are drawn up and/or to the left from their origin. Although the values of the **Line.Height** and **Line.Width** elements can be negative, both **Line.Top+Line.Height** and **Line.Left+Line.Width** MUST be non-negative valid sizes.

The following are the parent elements, attributes, and child elements of the **Line** element.

Parent elements
<a href="#">ReportItems</a>
<a href="#">CellContents</a>
<a href="#">CustomReportItem.AltReportItem</a>

Attributes
<a href="#">Line.Name</a>

Child elements
<a href="#">Line.Style</a>
<a href="#">Line.ActionInfo</a>
<a href="#">Line.Bookmark</a>
<a href="#">Line.CustomProperties</a>
<a href="#">Line.DataElementName</a>
<a href="#">Line.DataElementOutput</a>
<a href="#">Line.DocumentMapLabel</a>
Line.Height
Line.Left
<a href="#">Line.RepeatWith</a>
<a href="#">Line.ToolTip</a>
Line.Top
<a href="#">Line.Visibility</a>
Line.Width
<a href="#">Line.ZIndex</a>

The following is the XML Schema definition of the **Line** element.



```

<xsd:complexType name="LineType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.13.1 Line.Name

The **Line.Name** attribute specifies a unique identifier for a **line**. This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [UTR15] that is unique for all the report items within a report.

Following is the parent element of the **Line.Name** attribute.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.Name** attribute.

```

<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />

```

### 2.13.2 Line.Style

The **Line.Style** element specifies style information for a **line**. This element is optional and is of type [Style](#).

Following is the parent element of the **Line.Style** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.13.3 Line.ActionInfo

The **Line.ActionInfo** element is ignored.

Following is the parent element of the **Line.ActionInfo** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

In [RDL 2003/10](#) and [RDL 2005/01](#), the equivalent element of **Line.ActionInfo** is **Line.Action**, which is of type [Action](#).

### 2.13.4 Line.Bookmark

The **Line.Bookmark** element specifies a **bookmark** for a **line** that can be linked to by using an **action** with the [Action.BookmarkLink](#) element set. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Line.Bookmark** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.Bookmark** element.

```
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
```

### 2.13.5 Line.CustomProperties

The **Line.CustomProperties** element specifies custom information for a line that will be handed to a report rendering component. This element is optional and is of type [CustomProperties](#).

Following is the parent element of the **Line.CustomProperties** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

In [RDL 2003/10](#), the equivalent element of **Line.CustomProperties** is **Line.Custom**, which is of type [Custom](#).

### 2.13.6 Line.DataElementName

The **Line.DataElementName** element is ignored.

Following is the parent element of the **Line.DataElementName** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.13.7 Line.DataElementOutput

The **Line.DataElementOutput** element is ignored.

Following is the parent element of the **Line.DataElementOutput** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0" ">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">  
      <xsd:enumeration value="Output" />  
      <xsd:enumeration value="NoOutput" />  
      <xsd:enumeration value="ContentsOnly" />  
      <xsd:enumeration value="Auto" />  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:element>
```

### 2.13.8 Line.DocumentMapLabel

The **Line.DocumentMapLabel** element specifies a label to identify a line within the client UI in order to provide a user-friendly label for searching. This element is optional.

If the **Line.DocumentMapLabel** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. This element is ignored when used within a line that is within a [PageSection](#) or [PageHeaderFooter](#).

Following is the parent element of the **Line.DocumentMapLabel** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.DocumentMapLabel** element.

```
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
```

In [RDL 2003/10](#) and [RDL 2005/01](#), this element is called **Line.Label**. The following is the XML Schema definition of the **Line.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

### 2.13.9 Line.Height

The **Line.Height** element specifies the height of a line. This element is optional. If this element is present, its value MUST be an [RdlSize](#). This element MAY have a negative **RdlSize**. A "negative **RdlSize**" is an **RdlSize** in the format `<value><unit>` where *value* is a negative value.

If the **Line.Height** element is not present, its value is interpreted as the height of its container minus the value of the peer [Line.Top](#) element, if specified. If **Line.Top** is not specified, the value of **Line.Top** is Opt. This element MUST NOT have a size such that the sum of its size and that of the peer **Line.Top** element is negative.

Following is the parent element of the **Line.Height** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="0" />
```

### 2.13.10 Line.Left

The **Line.Left** element specifies the distance of a line from the left of the line's container. This element is optional. If this element is present, its value MUST be an [RdlSize](#). This element MAY have a negative **RdlSize**. This element MUST NOT have a size such that the sum of its size and that of the peer [Line.Width](#) element is negative.

Following is the parent element of the **Line.Left** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.Left** element.

```
<xsd:element name="Left" type="SizeType" minOccurs="0" />
```

### 2.13.11 Line.RepeatWith

The **Line.RepeatWith** element specifies the name of a **data region** in which a line is repeated if that data region spans multiple pages. The data region MUST be in the same [ReportItems](#) collection as this line. If the line is within a [PageSection](#) or [PageHeaderFooter](#), this element is ignored. The **Line.RepeatWith** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Following is the parent element of the **Line.RepeatWith** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.RepeatWith** element.

```
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
```

### 2.13.12 **Line.ToolTip**

The **Line.ToolTip** element is ignored.

Following is the parent element of the **Line.ToolTip** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

### 2.13.13 **Line.Top**

The **Line.Top** element specifies the distance of a line from the top of the line's container. This element is optional. If this element is present, its value **MUST** be an [RdlSize](#). This element can have a negative **RdlSize**. This element **MUST NOT** have a size such that the sum of its size and that of the peer [Line.Height](#) element is negative.

Following is the parent element of the **Line.Top** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.Top** element.

```
<xsd:element name="Top" type="SizeType" minOccurs="0" />
```

### 2.13.14 **Line.Visibility**

The **Line.Visibility** element specifies whether a line is hidden. This element is optional and is of type [Visibility](#).

Following is the parent element of the **Line.Visibility** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

### 2.13.15 Line.Width

The **Line.Width** element specifies the width of a line. This element is optional. If this element is present, its value MUST be an [RdISize](#). This element can have a negative **RdISize**.

If this element is not present, its value is interpreted as the width of the line's container (such as a **Rectangle** or **Body**) minus the value of the peer [Line.Left](#) element, if specified. If the **Line.Left** element is not specified, its value is 0.

The value of the **Line.Width** element MUST NOT be a size such that the sum of its size and that of the peer **Line.Left** element is negative.

Following is the parent element of the **Line.Width** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="0" />
```

### 2.13.16 Line.ZIndex

The **Line.ZIndex** element specifies the drawing order of a [Line](#) within its container. The **Line.ZIndex** element is optional. If this element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/21](#) section 3.3.17). If this element is not present, its value is interpreted as 0. The value of this element MUST be greater than or equal to 0 and less than or equal to 2147483647.

Following is the parent element of the **Line.ZIndex** element.

Parent elements
<a href="#">Line</a>

The following is the XML Schema definition of the **Line.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
```

## 2.14 Rectangle

The **Rectangle** element specifies properties for a **rectangle** that is to be drawn within a [Report](#).

The following are the parent elements, attributes, and child elements of the **Rectangle** element.

Parent elements
<a href="#">ReportItems</a>
<a href="#">CellContents</a>
<a href="#">CustomReportItem.AltReportItem</a>

Attributes
<a href="#">Rectangle.Name</a>

Child elements
<a href="#">Rectangle.Style</a>
<a href="#">Rectangle.ActionInfo</a>
<a href="#">Rectangle.Bookmark</a>
<a href="#">Rectangle.CustomProperties</a>
<a href="#">Rectangle.DataElementName</a>
<a href="#">Rectangle.DataElementOutput</a>
<a href="#">Rectangle.DocumentMapLabel</a>
<a href="#">Rectangle.Height</a>
<a href="#">Rectangle.Left</a>
<a href="#">Rectangle.RepeatWith</a>
<a href="#">Rectangle.ToolTip</a>
<a href="#">Rectangle.Top</a>
<a href="#">Rectangle.Visibility</a>
<a href="#">Rectangle.Width</a>
<a href="#">Rectangle.ZIndex</a>
<a href="#">Rectangle.KeepTogether</a>
<a href="#">Rectangle.LinkToChild</a>
<a href="#">Rectangle.OmitBorderOnPageBreak</a>
<a href="#">Rectangle.PageName</a>
<a href="#">Rectangle.PageBreak</a>
<a href="#">Rectangle.ReportItems</a>
<a href="#">Rectangle.PageBreakAtEnd</a>
<a href="#">Rectangle.PageBreakAtStart</a>

Following is the XML Schema definition of the **Rectangle** element in [RDL 2003/10](#) and [RDL 2005/01](#).

```
<xsd:complexType name="RectangleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0"/>
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

```

<xsd:element name="Left" type="SizeType" minOccurs="0" />
<xsd:element name="Height" type="SizeType" minOccurs="0" />
<xsd:element name="Width" type="SizeType" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
<xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0"/>
<xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output"/>
      <xsd:enumeration value="NoOutput"/>
      <xsd:enumeration value="ContentsOnly"/>
      <xsd:enumeration value="Auto"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

Following is the XML Schema definition of the **Rectangle** element in [RDL 2008/01](#).

```

<xsd:complexType name="RectangleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />

```



```

    <xsd:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>

```

Following is the XML Schema definition of the **Rectangle** element in [RDL 2010/01](#) and [RDL 2016/01](#).

```

<xsd:complexType name="RectangleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
    <xsd:element name="PageName" type="xsd:string" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.14.1 Rectangle.Name

The **Rectangle.Name** attribute specifies a unique identifier for a [Rectangle](#). This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique for all report items within a [Report](#).

Following is the parent element of the **Rectangle.Name** attribute.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.Name** attribute.

```

<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />

```

### 2.14.2 Rectangle.Style

The **Rectangle.Style** element specifies style properties for a [Rectangle](#). This element is optional and is of type [Style](#).

Following is the parent element of the **Rectangle.Style** element.

Parent elements
<a href="#">Rectangle</a>

The following is the XML Schema definition of the **Rectangle.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.14.3 Rectangle.ActionInfo

The **Rectangle.ActionInfo** element is ignored.

Following is the parent element of the **Rectangle.ActionInfo** element.

Parent elements
<a href="#">Rectangle</a>

The following is the XML Schema definition of the **Rectangle.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

In [RDL 2003/10](#) and [RDL 2005/01](#), the equivalent element of **Rectangle.ActionInfo** is **Rectangle.Action**, which is of type [Action](#).

### 2.14.4 Rectangle.Bookmark

The **Rectangle.Bookmark** element specifies a bookmark for a [Rectangle](#) that can be linked to via an **action** with the [Action.BookmarkLink](#) element set. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Rectangle.Bookmark** element.

Parent elements
<a href="#">Rectangle</a>

The following is the XML Schema definition of the **Rectangle.Bookmark** element.

```
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
```

### 2.14.5 Rectangle.CustomProperties

The **Rectangle.CustomProperties** element specifies custom information for a [Rectangle](#) that will be handed to a report rendering component. The **Rectangle.CustomProperties** element is optional. This element is of type [CustomProperties](#).

Following is the parent element of the **Rectangle.CustomProperties** element.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

In [RDL 2003/10](#), the equivalent element of **Rectangle.CustomProperties** is **Rectangle.Custom**, which is of type [Custom](#).

### 2.14.6 Rectangle.DataElementName

The **Rectangle.DataElementName** element specifies the name to use for the **data element** or the data attribute of a [Rectangle](#), depending on the setting of [Report.DataElementStyle](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as the **Name** attribute of the **rectangle**.

Following is the parent element of the **Rectangle.DataElementName** element.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.14.7 Rectangle.DataElementOutput

The **Rectangle.DataElementOutput** element specifies whether a [Rectangle](#) appears in a data (XML, CSV) rendering. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**Auto:** Specifies the default setting for how a **rectangle** appears in a **data rendering**, which is the same as "NoOutput" if the value of the [Visibility.Hidden](#) property for the rectangle is set to true and the [Visibility.ToggleItem](#) property is not specified. Otherwise, this value is the same as "ContentsOnly".

**Output:** Specifies that the rectangle appears in a data rendering output.

**NoOutput:** Specifies that the rectangle does not appear in a data rendering output.

**ContentsOnly:** Specifies that the rectangle does not appear in a data rendering output, but that its inner contents appear in a data rendering output.

If this element is not present, its value is interpreted as "Auto".

Following is the parent element of the **Rectangle.DataElementOutput** element.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.14.8 Rectangle.DocumentMapLabel

The **Rectangle.DocumentMapLabel** element specifies a label to identify a [Rectangle](#) within the client UI in order to provide a user-friendly label for searching. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The **Rectangle.DocumentMapLabel** element is ignored when used within a **rectangle** that is contained in a [PageSection](#) or [PageHeaderFooter](#).

Following is the parent element of the **Rectangle.DocumentMapLabel** element.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.DocumentMapLabel** element.

```
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
```

In [RDL 2003/10](#) and [RDL 2005/01](#), this element is called **Rectangle.Label**. The following is the XML Schema definition of the **Rectangle.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

### 2.14.9 Rectangle.Height

The **Rectangle.Height** element specifies the height of a [Rectangle](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as height of the rectangle's container minus the value of the peer [Rectangle.Top](#) element, if specified.

Following is the parent element of the **Rectangle.Height** element.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="0" />
```

#### 2.14.10 Rectangle.Left

The **Rectangle.Left** element specifies the distance of a [Rectangle](#) from the left of the rectangle's container. This element is optional. If this element is present, its value MUST be an [RdlSize](#).

Following is the parent element of the **Rectangle.Left** element.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.Left** element.

```
<xsd:element name="Left" type="SizeType" minOccurs="0" />
```

#### 2.14.11 Rectangle.RepeatWith

The **Rectangle.RepeatWith** element specifies the name of a **data region** in which a [Rectangle](#) is repeated if that data region spans multiple pages. The data region MUST be in the same ReportItems collection as the [Line](#). If this rectangle is within a [PageSection](#) or [PageHeaderFooter](#), the **Rectangle.RepeatWith** element is ignored.

This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Following is the parent element of the **Rectangle.RepeatWith** element.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.RepeatWith** element.

```
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
```

#### 2.14.12 Rectangle.ToolTip

The **Rectangle.ToolTip** element specifies the tooltip text for a [Rectangle](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **Rectangle.ToolTip** element is optional. The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Rectangle.ToolTip** element.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

### 2.14.13 Rectangle.Top

The **Rectangle.Top** element specifies the distance of a [Rectangle](#) from the top of the rectangle's container. This element is optional. If this element is present, its value MUST be an [RdlSize](#).

Following is the parent element of the **Rectangle.Top** element.

Parent elements
<a href="#">Rectangle</a>

The following is the XML Schema definition of the **Rectangle.Top** element.

```
<xsd:element name="Top" type="SizeType" minOccurs="0" />
```

### 2.14.14 Rectangle.Visibility

The **Rectangle.Visibility** element specifies whether a Rectangle is hidden. This element is optional and is of type [Visibility](#).

Following is the parent element of the **Rectangle.Visibility** element.

Parent elements
<a href="#">Rectangle</a>

The following is the XML Schema definition of the **Rectangle.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

### 2.14.15 Rectangle.Width

The **Rectangle.Width** element specifies the width of a [Rectangle](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as the width of the rectangle's container (such as a **Rectangle** or **Body**) minus the value of the peer [Rectangle.Left](#) element, if specified.

Following is the parent element of the **Rectangle.Width** element.

Parent elements
<a href="#">Rectangle</a>

The following is the XML Schema definition of the **Rectangle.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="0" />
```

### 2.14.16 **Rectangle.ZIndex**

The **Rectangle.ZIndex** element specifies the drawing order of a [Rectangle](#) within its container. This element is optional. If this element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17). If this element is not present, its value is interpreted as 0. The value of this element MUST be greater than or equal to 0 and less than or equal to 2147483647.

Following is the parent element of the **Rectangle.ZIndex** element.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.14.17 **Rectangle.KeepTogether**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Rectangle.KeepTogether** element specifies whether the entire contents of a [Rectangle](#) is kept together on one page if possible. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Rectangle.KeepTogether** element.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.KeepTogether** element.

```
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
```

### 2.14.18 **Rectangle.LinkToChild**

The **Rectangle.LinkToChild** element specifies the name of a report item that is contained within a [Rectangle](#) that is the target location for the peer [Rectangle.DocumentMapLabel](#) element. The **Rectangle.LinkToChild** element is ignored if the peer **Rectangle.DocumentMapLabel** element is not present.

The **Rectangle.LinkToChild** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, and if **Rectangle.DocumentMapLabel** is present, then target location for the peer element **Rectangle.DocumentMapLabel** is the rectangle itself.

Following is the parent element of the **Rectangle.LinkToChild** element.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.LinkToChild** element.

```
<xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
```

### 2.14.19 **Rectangle.OmitBorderOnPageBreak**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Rectangle.OmitBorderOnPageBreak** element specifies whether borders do not appear at locations where a [Rectangle](#) spans multiple pages. This element also specifies whether [BackgroundImage](#) instances within the rectangle continue rather than restart after a page break.

The **Rectangle.OmitBorderOnPageBreak** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Rectangle.OmitBorderOnPageBreak** element.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.OmitBorderOnPageBreak** element.

```
<xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
```

### 2.14.20 **Rectangle.PageBreak**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Rectangle.PageBreak** element specifies page break properties for a [Rectangle](#). This element is optional and is of type [PageBreak](#).

Following is the parent element of the **Rectangle.PageBreak** element.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.PageBreak** element.

```
<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
```

### 2.14.21 **Rectangle.ReportItems**

The **Rectangle.ReportItems** element specifies **report items** that are to be contained within the bounds of a [Rectangle](#). This element is optional and is of type [ReportItems](#).

Following is the parent element of the **Rectangle.ReportItems** element.

Parent elements
Rectangle

The following is the XML Schema definition of the **Rectangle.ReportItems** element.



```
<xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
```

### 2.14.22 **Rectangle.PageBreakAtEnd**

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Rectangle.PageBreakAtEnd** element indicates that the **renderer** places a page break at the end of the [Rectangle](#). This element is optional. The value of this element MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Rectangle.PageBreakAtEnd** element.

Parent elements
<a href="#">Rectangle</a>

The following is the XML Schema definition of the **Rectangle.PageBreakAtEnd** element.

```
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
```

### 2.14.23 **Rectangle.PageBreakAtStart**

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Rectangle.PageBreakAtStart** element indicates that the **renderer** places a page break at the start of a [Rectangle](#). This element is optional. The value of this element MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Rectangle.PageBreakAtStart** element.

Parent elements
<a href="#">Rectangle</a>

The following is the XML Schema definition of the **Rectangle.PageBreakAtStart** element.

```
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
```

### 2.14.24 **Rectangle.PageName**

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Rectangle.PageName** element specifies the value to use for the name of a paginated page. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as NULL.

Following is the parent element of the **Rectangle.PageName** element.

Parent elements
<a href="#">Rectangle</a>

The following is the XML Schema definition of the **Rectangle.PageName** element.

```
<xsd:element name="PageName" type="xsd:string" minOccurs="0" />
```

## 2.15 Subreport

The **Subreport** element specifies a subreport to be processed and rendered inline in the [Report](#).

Failure to process a subreport results in a **text box** that replaces the subreport and that contains the string "Error: Subreport could not be shown". The [Subreport.Style](#) information applies to the text box.

Subreports that are hidden and cannot be made visible via a [Visibility.ToggleItem](#) are not executed.

The following report properties do not apply when a report is used as a subreport:

- [Report.Description](#)
- [Report.Author](#)
- [Report.AutoRefresh](#)
- [Report.Width](#)
- [Report.Page](#)
- [Report.DataTransform](#)
- [Report.DataSchema](#)
- [ReportSection.Page](#)

The following are the parent elements, attributes, and child elements of the **Subreport** element.

Parent elements
<a href="#">ReportItems</a>
<a href="#">CellContents</a>
<a href="#">CustomReportItem.AltReportItem</a>

Attributes
<a href="#">Subreport.Name</a>

Child elements
Subreport.Style
<a href="#">Subreport.ActionInfo</a>
<a href="#">Subreport.Bookmark</a>
<a href="#">Subreport.CustomProperties</a>
<a href="#">Subreport.DataElementName</a>

Child elements
<a href="#">Subreport.DataElementOutput</a>
<a href="#">Subreport.DocumentMapLabel</a>
<a href="#">Subreport.Height</a>
<a href="#">Subreport.Left</a>
<a href="#">Subreport.LinkToChild</a>
<a href="#">Subreport.RepeatWith</a>
<a href="#">Subreport.ToolTip</a>
<a href="#">Subreport.Top</a>
<a href="#">Subreport.Visibility</a>
<a href="#">Subreport.Width</a>
<a href="#">Subreport.ZIndex</a>
<a href="#">Subreport.KeepTogether</a>
<a href="#">Subreport.MergeTransactions</a>
<a href="#">Subreport.NoRowsMessage</a>
<a href="#">Subreport.OmitBorderOnPageBreak</a>
<a href="#">Subreport.Parameters</a>
<a href="#">Subreport.ReportName</a>
<a href="#">Subreport.NoRows</a>

The following is the XML Schema definition of the **Subreport** element in [RDL 2003/10](#) and [RDL 2005/01](#).

```

<xsd:complexType name="SubreportType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0"/>
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0"/>
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0"/>
    <xsd:element name="ReportName" type="xsd:string" />
    <xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
    <xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
    <xsd:element name="MergeTransactions" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">

```

```

        <xsd:enumeration value="Output"/>
        <xsd:enumeration value="NoOutput"/>
        <xsd:enumeration value="ContentsOnly"/>
        <xsd:enumeration value="Auto"/>
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **Subreport** element in [RDL 2008/01](#).

```

<xsd:complexType name="SubreportType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="ReportName" type="xsd:string" />
    <xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
    <xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
    <xsd:element name="MergeTransactions" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **Subreport** element in [RDL 2010/01](#) and [RDL 2016/01](#).

```

<xsd:complexType name="SubreportType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

<xsd:element name="Width" type="SizeType" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
<xsd:element name="ReportName" type="xsd:string" />
<xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
<xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
<xsd:element name="MergeTransactions" type="xsd:boolean" minOccurs="0" />
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
<xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.15.1 Subreport.Name

The **Subreport.Name** attribute specifies a unique identifier for a [Subreport](#) definition. This attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier [UTR15]** that is unique for all report items within a [Report](#).

Following is the parent element of the **Subreport.Name** attribute.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.15.2 Subreport.Style

The **Subreport.Style** element specifies style properties for a [Subreport](#). This element is optional. This element is of type [Style](#).

Following is the parent element of the **Subreport.Style** element.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0"/>
```

### 2.15.3 Subreport.ActionInfo

The **Subreport.ActionInfo** element is ignored.

The following is the XML Schema definition of the **Subreport.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

In [RDL 2003/10](#) and [RDL 2005/01](#), the equivalent element of **Subreport.ActionInfo** is **Subreport.Action**, which is of type [Action](#).

### 2.15.4 Subreport.Bookmark

The **Subreport.Bookmark** element specifies a bookmark that can be linked to via a bookmark action for a [Subreport](#). The **Subreport.Bookmark** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Subreport.Bookmark** element.

Parent elements
<a href="#">Subreport</a>

The following is the XML Schema definition of the **Subreport.Bookmark** element.

```
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
```

### 2.15.5 Subreport.CustomProperties

The **Subreport.CustomProperties** element specifies custom information for a subreport that will be made available to a report rendering component. This element is optional. This element is of type [CustomProperties](#).

Following is the parent element of the **Subreport.CustomProperties** element.

Parent elements
<a href="#">Subreport</a>

The following is the XML Schema definition of the **Subreport.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0"/>
```

In [RDL 2003/10](#), the equivalent element of **Subreport.CustomProperties** is **Subreport.Custom**, which is of type [Custom](#).

### 2.15.6 Subreport.DataElementName

The **Subreport.DataElementName** element specifies the name to use for the **data element** or attribute of a [Subreport](#). This element is optional. If this element is present, it MUST be a **CLS-**

**compliant identifier** [\[UTR15\]](#). If this element is not present, its value defaults to the name of the subreport.

Following is the parent element of the **Subreport.DataElementName** element.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.15.7 Subreport.DataElementOutput

The **Subreport.DataElementOutput** element indicates how a [Subreport](#) appears in a **data rendering**. This element is optional. If this element is not present, its value defaults to "Auto". If this element is present, its value MUST be one of the following:

**Auto** (Default): MUST behave as "NoOutput" if the visibility of the subreport is statically hidden. The subreport is statically hidden when the subreport's [Visibility.Hidden](#) property is set to true and the [Visibility.ToggleItem](#) property is not specified, or if the subreport is within a [TablixMember](#) that has groupings that have the **Visibility.Hidden** property set to non-expression true. Otherwise, "Auto" MUST behave as "Output".

**Output:** Indicates that the subreport appears in a data rendering output.

**NoOutput:** Indicates that the subreport does not appear in a data rendering output.

Following is the parent element of the **Subreport.DataElementOutput** element.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0" >
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.15.8 Subreport.DocumentMapLabel

The **Subreport.DocumentMapLabel** element specifies a label to identify a [Subreport](#) within the client UI in order to provide a user-friendly label for searching. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If the expression evaluates to null, an item for the subreport MUST NOT be added to the **document map**. The **Subreport.DocumentMapLabel** element MUST NOT be used within a subreport that is contained within a [PageSection](#) or a [PageHeaderFooter](#).

Following is the parent element of the **Subreport.DocumentMapLabel** element.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.DocumentMapLabel** element.

```
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
```

In [RDL 2003/10](#) and [RDL 2005/01](#), the **Subreport.DocumentMapLabel** element is called **Subreport.Label**. The following is the XML Schema definition of the **Subreport.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

### 2.15.9 Subreport.Height

The **Subreport.Height** element specifies the height of a [Subreport](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as the height of the subreport's container, such as [Rectangle](#) or [Body](#), minus the value of the [Subreport.Top](#) element, if specified.

Following is the parent element of the **Subreport.Height** element.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="0" />
```

### 2.15.10 Subreport.Left

The **Subreport.Left** element specifies the distance of the subreport from the left of the subreport's container, such as a [Rectangle](#) or [Body](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#).

Following is the parent element of the **Subreport.Left** element.

Parent elements
<a href="#">Subreport</a>

The following is the XML Schema definition of the **Subreport.Left** element.

```
<xsd:element name="Left" type="SizeType" minOccurs="0" />
```

### 2.15.11 Subreport.LinkToChild

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)



The **Subreport.LinkToChild** element is ignored.

Following is the parent element of the **Subreport.LinkToChild** element.

Parent elements
<a href="#">Subreport</a>

The following is the XML Schema definition of the **Subreport.LinkToChild** element.

```
<xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
```

### 2.15.12 Subreport.RepeatWith

The **Subreport.RepeatWith** element MUST NOT be specified.

Following is the parent element of the **Subreport.RepeatWith** element.

Parent elements
<a href="#">Subreport</a>

The following is the XML Schema definition of the **Subreport.RepeatWith** element.

```
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
```

### 2.15.13 Subreport.ToolTip

The **Subreport.ToolTip** element specifies the tooltip text for a [Subreport](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **Subreport.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Subreport.ToolTip** element.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

### 2.15.14 Subreport.Top

The **Subreport.Top** element specifies the distance of the subreport from the top of the subreport's container, such as a [Rectangle](#) or [Body](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#).

Following is the parent element of the **Subreport.Top** element.

<b>Parent elements</b>
<a href="#">Tablix</a>

The following is the XML Schema definition of the **Subreport.Top** element.

```
<xsd:element name="Top" type="SizeType" minOccurs="0" />
```

### 2.15.15 Subreport.Visibility

The **Subreport.Visibility** element specifies whether a [Subreport](#) is hidden or toggled by a [Textbox](#). This element is optional. This element is of type [Visibility](#).

Following is the parent element of the **Subreport.Visibility** element.

<b>Parent elements</b>
<a href="#">Subreport</a>

The following is the XML Schema definition of the **Subreport.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0"/>
```

### 2.15.16 Subreport.Width

The **Subreport.Width** element specifies the width of a [Subreport](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as the width of the subreport's container (such as a [Rectangle](#) or [Body](#)) minus the value of the [Subreport.Left](#) element, if specified.

Following is the parent element of the **Subreport.Width** element.

<b>Parent elements</b>
<a href="#">Subreport</a>

The following is the XML Schema definition of the **Subreport.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="0" />
```

### 2.15.17 Subreport.ZIndex

The **Subreport.ZIndex** element specifies the drawing order of a [Subreport](#) within its container. This element is optional. If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17). If this element is not present, its value is interpreted as 0. The value of this element MUST be greater than or equal to 0 and less than or equal to 2147483647, if a value is specified.

Following is the parent element of the **Subreport.ZIndex** element.

<b>Parent elements</b>
<a href="#">Subreport</a>

The following is the XML Schema definition of the **Subreport.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.15.18 Subreport.KeepTogether

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Subreport.KeepTogether** element indicates that the entire [Subreport](#) is kept together on one [Page](#), if possible. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Subreport.KeepTogether** element.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.KeepTogether** element.

```
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
```

### 2.15.19 Subreport.MergeTransactions

The **Subreport.MergeTransactions** element indicates that the transactions in a [Subreport](#) MUST be merged with transactions in the parent [Report](#) (that is, the transactions are merged into a single transaction for the entire report), if the data sources use the same connection.

The **Subreport.MergeTransactions** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Subreport.MergeTransactions** element.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.MergeTransactions** element.

```
<xsd:element name="MergeTransactions" type="xsd:boolean" minOccurs="0" />
```

### 2.15.20 Subreport.NoRowsMessage

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Subreport.NoRowsMessage** element indicates the message to display in the place of a [Subreport](#) when no rows of data are available in any [DataSet](#) that is used in the body of the subreport. This element is optional. If this element is present, its value MUST either be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. [Style](#) information MUST apply to this text.

Following is the parent element of the **Subreport.NoRowsMessage** element.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.NoRowsMessage** element.

```
<xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
```

### 2.15.21 Subreport.OmitBorderOnPageBreak

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Subreport.OmitBorderOnPageBreak** element specifies whether borders appear at locations where a [Subreport](#) spans multiple pages. This element also specifies whether [BackgroundImage](#) instances within the subreport continue rather than restart after a page break.

This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Subreport.OmitBorderOnPageBreak** element.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.OmitBorderOnPageBreak** element.

```
<xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
```

### 2.15.22 Subreport.Parameters

The **Subreport.Parameters** element specifies the parameters to be evaluated and passed to the [Subreport](#). This element is optional. This element is of type [Parameters](#).

Following is the parent element of the **Subreport.Parameters** element.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.Parameters** element.

```
<xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
```

### 2.15.23 Subreport.ReportName

The **Subreport.ReportName** element specifies the location of the **report definition** to use for the [Subreport](#). This element MUST be specified. The location MUST specify, as a non-empty string (ignoring white space), the **absolute path** (such as "/salesreports/orderdetails") or the **relative path** (such as "orderdetails") to a subreport on the same server. Relative paths start in the same folder as the main report, not the subreport.

Following is the parent element of the **Subreport.ReportName** element.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.ReportName** element.

```
<xsd:element name="ReportName" type="xsd:string" />
```

### 2.15.24 Subreport.NoRows

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Subreport.NoRows** element indicates the message to display in the place of a [Subreport](#) when no rows of data are available in any [DataSet](#) that is used in the body of the subreport. This element is optional. If this element is specified, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. [Style](#) information MUST apply to this text.

Following is the parent element of the **Subreport.NoRows** element.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.NoRows** element.

```
<xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
```

### 2.16 Textbox

The **Textbox** element specifies properties for a **text box** in a report.

Following are the parent elements, attributes, and child elements of the **Textbox** element.

Parent elements
<a href="#">ReportItems</a>
<a href="#">CellContents</a>
<a href="#">CustomReportItem.AltReportItem</a>

Attributes
<a href="#">Textbox.Name</a>

Child elements
<a href="#">Textbox.Style</a>
<a href="#">Textbox.ActionInfo</a>

Child elements
<a href="#">Textbox.Bookmark</a>
<a href="#">Textbox.CustomProperties</a>
<a href="#">Textbox.DataElementName</a>
<a href="#">Textbox.DataElementOutput</a>
<a href="#">Textbox.DocumentMapLabel</a>
<a href="#">Textbox.Height</a>
<a href="#">Textbox.Left</a>
<a href="#">Textbox.RepeatWith</a>
<a href="#">Textbox.ToolTip</a>
<a href="#">Textbox.Top</a>
<a href="#">Textbox.Visibility</a>
<a href="#">Textbox.Width</a>
<a href="#">Textbox.ZIndex</a>
<a href="#">Textbox.CanGrow</a>
<a href="#">Textbox.CanShrink</a>
<a href="#">Textbox.DataElementStyle</a>
<a href="#">Textbox.HideDuplicates</a>
<a href="#">Textbox.KeepTogether</a>
<a href="#">Textbox.Paragraphs</a>
<a href="#">Textbox.ToggleImage</a>
<a href="#">Textbox.UserSort</a>
<a href="#">Textbox.Value</a>

**Applies to [RDL 2011/01](#)**

Child elements
<a href="#">Textbox.CanScrollVertically</a>

The following is the XML Schema definition of the **Textbox** element in [RDL 2003/10](#) and [RDL 2005/01](#).

```

<xsd:complexType name="TextboxType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>

```

```

<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
<xsd:element name="Value" type="xsd:string" minOccurs="1" />
<xsd:element name="CanGrow" type="xsd:boolean" minOccurs="0" />
<xsd:element name="CanShrink" type="xsd:boolean" minOccurs="0" />
<xsd:element name="HideDuplicates" type="xsd:string" minOccurs="0" />
<xsd:element name="ToggleImage" type="ToggleImageType" minOccurs="0" />
<xsd:element name="UserSort" type="UserSortType" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="DataElementStyle" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Auto" />
      <xsd:enumeration value="AttributeNormal" />
      <xsd:enumeration value="ElementNormal" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **Textbox** element in [RDL 2008/01](#).

```

<xsd:complexType name="TextboxType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="Paragraphs" type="ParagraphsType" minOccurs="1" />
    <xsd:element name="CanGrow" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CanShrink" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="HideDuplicates" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToggleImage" type="ToggleImageType" minOccurs="0" />
    <xsd:element name="UserSort" type="UserSortType" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>

```

```

        <xsd:enumeration value="ContentsOnly" />
        <xsd:enumeration value="Auto" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="DataElementStyle" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Auto" />
            <xsd:enumeration value="Attribute" />
            <xsd:enumeration value="Element" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **Textbox** element in [RDL 2010/01](#) and [RDL 2016/01](#).

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```

<xsd:complexType name="TextboxType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:element name="Top" type="SizeType" minOccurs="0" />
        <xsd:element name="Left" type="SizeType" minOccurs="0" />
        <xsd:element name="Height" type="SizeType" minOccurs="0" />
        <xsd:element name="Width" type="SizeType" minOccurs="0" />
        <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
        <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
        <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
        <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
        <xsd:element name="Paragraphs" type="ParagraphsType" minOccurs="1" />
        <xsd:element name="CanGrow" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="CanShrink" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="HideDuplicates" type="xsd:string" minOccurs="0" />
        <xsd:element name="ToggleImage" type="ToggleImageType" minOccurs="0" />
        <xsd:element name="UserSort" type="UserSortType" minOccurs="0" />
        <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
        <xsd:element name="DataElementOutput" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Output" />
                    <xsd:enumeration value="NoOutput" />
                    <xsd:enumeration value="ContentsOnly" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="DataElementStyle" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Auto" />
                    <xsd:enumeration value="Attribute" />
                    <xsd:enumeration value="Element" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:choice>

```



```

        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.16.1 Textbox.Name

The **Textbox.Name** attribute specifies a unique identifier for a [Textbox](#). This attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [UTR15] that is unique for all report items within a [Report](#).

Following is the parent element of the **Textbox.Name** attribute.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.16.2 Textbox.Style

The **Textbox.Style** element specifies style information for a [Textbox](#). This element is optional. This element is of type [Style](#).

Following is the parent element of the **Textbox.Style** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.16.3 Textbox.ActionInfo

The **Textbox.ActionInfo** element specifies one or more **actions** (such as a hyperlink) associated with a [Textbox](#). This element is optional. This element is of type [ActionInfo](#).

Following is the parent element of the **Textbox.ActionInfo** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

In [RDL 2003/10](#) and [RDL 2005/01](#), the equivalent element of **Textbox.ActionInfo** is **Textbox.Action**, which is of type [Action](#).

#### 2.16.4 Textbox.Bookmark

The **Textbox.Bookmark** element specifies a bookmark that can be linked to via a bookmark action to a [Textbox](#). This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Textbox.Bookmark** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.Bookmark** element.

```
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
```

#### 2.16.5 Textbox.CanScrollVertically

*Applies to [RDL 2011/01](#)*

The **Textbox.CanScrollVertically** element specifies whether a [Textbox](#) has a vertical scrollbar. The **Textbox.CanScrollVertically** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Textbox.CanScrollVertically** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.CanScrollVertically** element.

```
<xsd:element name="CanScrollVertically" type="xsd:boolean" minOccurs="0" />
```

#### 2.16.6 Textbox.CustomProperties

The **Textbox.CustomProperties** element specifies custom information for a [Textbox](#) that will be handed to a report rendering component. This element is optional. This element is of type [CustomProperties](#).

Following is the parent element of the **Textbox.CustomProperties** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

In [RDL 2003/10](#), the equivalent element of **Textbox.CustomProperties** is **Textbox.Custom**, which is of type [Custom](#).

### 2.16.7 Textbox.DataElementName

The **Textbox.DataElementName** element specifies the name to use for the **data element** or attribute of a [Textbox](#). The **Textbox.DataElementName** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as the [Textbox.Name](#) attribute of the text box.

Following is the parent element of the **Textbox.DataElementName** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.16.8 Textbox.DataElementOutput

The **Textbox.DataElementOutput** element specifies whether a [Textbox](#) appears in a data (XML, CSV) rendering. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). If this element is not present, its value is interpreted as "Auto". The value of this element MUST be one of the following:

**Auto:** Specifies the default setting for how the text box will appear in a **data rendering**, which MUST be the same as "NoOutput" if [Visibility.Hidden](#) for the text box is set to true and [Visibility.ToggleItem](#) is not set. Otherwise, this value is interpreted as "Output".

**Output:** Specifies that the text box appears in a data rendering output.

**NoOutput:** Specifies that the text box does not appear in a data rendering output.

Following is the parent element of the **Textbox.DataElementOutput** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0" ">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">  
      <xsd:enumeration value="Output" />  
      <xsd:enumeration value="NoOutput" />  
      <xsd:enumeration value="ContentsOnly" />  
      <xsd:enumeration value="Auto" />  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:element>
```

## 2.16.9 Textbox.DocumentMapLabel

The **Textbox.DocumentMapLabel** element specifies a label to identify a [Textbox](#) within the client UI in order to provide a user-friendly label for searching. This element is optional. If the **Textbox.DocumentMapLabel** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. This element is ignored when used within a **text box** that is contained within a [PageSection](#) or [PageHeaderFooter](#).

Following is the parent element of the **Textbox.DocumentMapLabel** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.DocumentMapLabel** element.

```
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
```

In [RDL 2003/10](#) and [RDL 2005/01](#), this element is called **Textbox.Label**. The following is the XML Schema definition of the **Textbox.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

## 2.16.10 Textbox.Height

The **Textbox.Height** element specifies the height of a [Textbox](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as the height of the text box's container, such as [Rectangle](#) or [Body](#), minus the value of the [Textbox.Top](#) element, if specified.

Following is the parent element of the **Textbox.Height** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="0" />
```

## 2.16.11 Textbox.Left

The **Textbox.Left** element specifies the distance of a [Textbox](#) from the left of the text box's container, such as a [Rectangle](#) or [Body](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#).

Following is the parent element of the **Textbox.Left** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.Left** element.

```
<xsd:element name="Left" type="SizeType" minOccurs="0" />
```

### 2.16.12 Textbox.RepeatWith

The **Textbox.RepeatWith** element specifies the name of a **data region** in which a [Textbox](#) is repeated with if the data region spans multiple pages. The data region **MUST** be in the same container as the text box. If the text box is within a [PageSection](#) or [PageHeaderFooter](#), this element is ignored.

This element is optional. If this element is present, its value **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Following is the parent element of the **Textbox.RepeatWith** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.RepeatWith** element.

```
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
```

### 2.16.13 Textbox.ToolTip

The **Textbox.ToolTip** element specifies a textual label for a [Textbox](#) that is used for things such as specifying **title** and **alt** attributes in HTML reports. The **alt** attribute can be used to render alternative text (alt text). The **Textbox.ToolTip** element is optional. If this element is present, its value **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Textbox.ToolTip** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

### 2.16.14 Textbox.Top

The **Textbox.Top** element specifies the distance of a [Textbox](#) from the top of the text box's container, such as a [Rectangle](#) or [Body](#). This element is optional. If this element is present, its value **MUST** be an [RdlSize](#).

Following is the parent element of the **Textbox.Top** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.Top** element.

```
<xsd:element name="Top" type="SizeType" minOccurs="0" />
```

### 2.16.15 Textbox.Visibility

The **Textbox.Visibility** element specifies whether a [Textbox](#) is hidden. This element is optional. This element is of type [Visibility](#).

Following is the parent element of the **Textbox.Visibility** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

### 2.16.16 Textbox.Width

The **Textbox.Width** element specifies the width of a [Textbox](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#). If the **Textbox.Width** element is not present, its value is interpreted as the width of the text box's container, such as [Rectangle](#) or [Body](#), minus the value of the [Textbox.Left](#) element, if specified.

Following is the parent element of the **Textbox.Width** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="0" />
```

### 2.16.17 Textbox.ZIndex

The **Textbox.ZIndex** element specifies the drawing order of a [Textbox](#) within its container. This element is optional. If this element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/21](#) section 3.3.17). If the **Textbox.ZIndex** element is not present, its value is interpreted as 0. The value of this element MUST be greater than or equal to 0 and less than or equal to 2147483647.

Following is the parent element of the **Textbox.ZIndex** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.16.18 Textbox.CanGrow

The **Textbox.CanGrow** element specifies whether the height of a [Textbox](#) can increase to match its contents. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Textbox.CanGrow** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.CanGrow** element.

```
<xsd:element name="CanGrow" type="xsd:boolean" minOccurs="0" />
```

### 2.16.19 Textbox.CanShrink

The **Textbox.CanShrink** element specifies whether the height of a [Textbox](#) can decrease to match its contents. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Textbox.CanShrink** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.CanShrink** element.

```
<xsd:element name="CanShrink" type="xsd:boolean" minOccurs="0" />
```

### 2.16.20 Textbox.DataElementStyle

The **Textbox.DataElementStyle** element specifies whether all [TextRun.Value](#) elements or all [Textbox.Value](#) elements, when rendered out to an XML data **renderer**, are rendered as attributes or as elements. This element is optional. If the **Textbox.DataElementStyle** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). The **String** MUST be one of the following:

**Auto:** Specifies that this value MUST be the same as [Report.DataElementStyle](#).

**Attribute:** Specifies that **TextRun.Value** elements or the **Textbox.Value** element within this text box are all rendered as attributes. This value was formerly "AttributeNormal" in [RDL 2003/10](#) and [RDL 2005/01](#).

**Element:** Specifies that **TextRun.Value** elements or the **TextRun.Value** element within this text box are all rendered as elements. This value was formerly "ElementNormal" in [RDL 2003/10](#) and [RDL 2005/01](#).

If the **Textbox.DataElementStyle** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **Textbox.DataElementStyle** element.

<b>Parent elements</b>
------------------------

<a href="#">Textbox</a>
-------------------------

The following is the XML Schema definition of the **Textbox.DataElementStyle** element in RDL 2003/10 and RDL 2005/01.

```
<xsd:element name="DataElementStyle" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Auto" />
      <xsd:enumeration value="AttributeNormal" />
      <xsd:enumeration value="ElementNormal" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

The following is the XML Schema definition of the **Textbox.DataElementStyle** element in [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#).

```
<xsd:element name="DataElementStyle" minOccurs="0" ">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Auto" />
      <xsd:enumeration value="Attribute" />
      <xsd:enumeration value="Element" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.16.21 Textbox.HideDuplicates

The **Textbox.HideDuplicates** element specifies whether duplicate text is displayed within a [Textbox](#) for a containing group or [DataSet](#). If the text box lies within a **data region**, and if the text box contains duplicate values for a group or **dataset**, only the text within the text box MUST be hidden, and the border and background of the text box MUST be shown. Separate group instances or dataset row data from previous report pages is ignored as potential duplicates by this element.

In [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#), if the text box contains more than one [TextRun](#), the **Textbox.HideDuplicates** element is ignored.

The **Textbox.HideDuplicates** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be a **String** that evaluates to the name of a containing group or dataset.

Following is the parent element of the **Textbox.HideDuplicates** element.

<b>Parent elements</b>
------------------------

<a href="#">Textbox</a>
-------------------------

The following is the XML Schema definition of the **Textbox.HideDuplicates** element.

```
<xsd:element name="HideDuplicates" type="xsd:string" minOccurs="0" />
```



## 2.16.22 Textbox.KeepTogether

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Textbox.KeepTogether** element specifies whether the entire contents of a [Textbox](#) will be kept together on one [Page](#) if possible. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Textbox.KeepTogether** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.KeepTogether** element.

```
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
```

## 2.16.23 Textbox.Paragraphs

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Textbox.Paragraphs** element specifies a collection of [Paragraph](#) elements within a [Textbox](#). This element is of type [Paragraphs](#). This element MUST be specified.

Following is the parent element of the **Textbox.Paragraphs** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.Paragraphs** element.

```
<xsd:element name="Paragraphs" type="ParagraphsType" minOccurs="1" />
```

## 2.16.24 Textbox.ToggleImage

The **Textbox.ToggleImage** element specifies the initial state (+/-) of a **toggle image** if one is displayed as part of a [Textbox](#). This element is optional and of type [ToggleImage](#).

Following is the parent element of the **Textbox.ToggleImage** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.ToggleImage** element.

```
<xsd:element name="ToggleImage" type="ToggleImageType" minOccurs="0" />
```

### 2.16.25 Textbox.UserSort

Applies to [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Textbox.UserSort** element specifies an end-user sort control that is displayed as part of a [Textbox](#) within the UI. This element is optional. This element is of type [UserSort](#).

Following is the parent element of the **Textbox.UserSort** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.UserSort** element.

```
<xsd:element name="UserSort" type="UserSortType" minOccurs="0" />
```

### 2.16.26 Textbox.Value

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **Textbox.Value** element specifies a value or an expression that is displayed for a [Textbox](#).

Following is the parent element of the **Textbox.Value** element.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.Value** element.

```
<xsd:element name="Value" type="xsd:string" />
```

## 2.17 Paragraphs

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Paragraphs** element specifies a collection of [Paragraph](#) elements. This element MUST contain at least one [Paragraphs.Paragraph](#) element.

The following are the parent elements and child elements of the **Paragraphs** element.

Parent elements
<a href="#">Textbox</a>

Child elements
Paragraphs.Paragraph

The following is the XML Schema definition of the **Paragraphs** element in RDL 2008/01.

```
<xsd:complexType name="ParagraphsType">
```

```

<xsd:sequence>
  <xsd:element name="Paragraph" type="ParagraphType" minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **Paragraphs** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="ParagraphsType">
  <xsd:sequence>
    <xsd:element name="Paragraph" type="ParagraphType" minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.17.1 Paragraphs.Paragraph

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Paragraphs.Paragraph** element specifies properties for a paragraph of text. This element MUST be specified at least once within a [Paragraphs](#) collection. This element is of type [Paragraph](#).

Following is the parent element of the **Paragraphs.Paragraph** element.

Parent elements
<a href="#">Paragraphs</a>

The following is the XML Schema definition of the **Paragraphs.Paragraph** element.

```

<xsd:element name="Paragraph" type="ParagraphType" minOccurs="1" maxOccurs="unbounded"/>

```

### 2.18 Paragraph

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Paragraph** element specifies the layout properties of a paragraph of text within a [Textbox](#) and contains a collection of [TextRun](#) elements. This element MUST be specified at least once within a [Paragraphs](#) collection.

The following are the parent and child elements of the **Paragraph** element.

Parent elements
<a href="#">Paragraphs</a>

Child elements
<a href="#">Paragraph.HangingIndent</a>
<a href="#">Paragraph.LeftIndent</a>
<a href="#">Paragraph.ListLevel</a>

Child elements
<a href="#">Paragraph.ListStyle</a>
<a href="#">Paragraph.RightIndent</a>
<a href="#">Paragraph.SpaceAfter</a>
<a href="#">Paragraph.SpaceBefore</a>
<a href="#">Paragraph.Style</a>
<a href="#">Paragraph.TextRuns</a>

The following is the XML Schema definition of the **Paragraph** element in RDL 2008/01.

```
<xsd:complexType name="ParagraphType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TextRuns" type="TextRunsType" minOccurs="1" />
    <xsd:element name="LeftIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="HangingIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="ListStyle" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Bulleted" />
          <xsd:enumeration value="Numbered" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="ListLevel" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="SpaceBefore" type="xsd:string" minOccurs="0" />
    <xsd:element name="SpaceAfter" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **Paragraph** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ParagraphType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TextRuns" type="TextRunsType" minOccurs="1" />
    <xsd:element name="LeftIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="HangingIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="ListStyle" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Bulleted" />
          <xsd:enumeration value="Numbered" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="ListLevel" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="SpaceBefore" type="xsd:string" minOccurs="0" />
    <xsd:element name="SpaceAfter" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

<xsd:complexType>

### 2.18.1 Paragraph.HangingIndent

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Paragraph.HangingIndent** element specifies the indentation of the first line of text within a [Paragraph](#) relative to the peer [Paragraph.LeftIndent](#) element of the paragraph. The **Paragraph.HangingIndent** element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**.

Following is the parent element of the **Paragraph.HangingIndent** element.

Parent elements
Paragraph

The following is the XML Schema definition of the **Paragraph.HangingIndent** element.

```
<xsd:element name="HangingIndent" type="xsd:string" minOccurs="0" />
```

### 2.18.2 Paragraph.LeftIndent

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Paragraph.LeftIndent** element specifies the indentation for text within a [Paragraph](#) from the left edge of its containing [Textbox](#). This element is optional. If the **Paragraph.LeftIndent** element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**.

Following is the parent element of the **Paragraph.LeftIndent** element.

Parent elements
Paragraph

The following is the XML Schema definition of the **Paragraph.LeftIndent** element.

```
<xsd:element name="LeftIndent" type="xsd:string" minOccurs="0" />
```

### 2.18.3 Paragraph.ListLevel

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Paragraph.ListLevel** element specifies the numbering style and/or indentation level of a [Paragraph](#) within a [Paragraphs](#) collection. This element is optional.

If the **Paragraph.ListLevel** element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) that is greater than or equal to 0. [<6>](#) If this element is not present, its value is interpreted as "0". The value of this element MUST be greater than or equal to 0 and less than or equal to 2147483647.

If the peer [Paragraph.ListStyle](#) element is set to "None", the **Paragraph.ListLevel** element MUST indent only the paragraph. If the peer **Paragraph.ListStyle** element is set to "Bulleted" or "Numbered", the **Paragraph.ListLevel** element MUST also specify the bulleting or numbering style.

Following is the parent element of the **Paragraph.ListLevel** element.

Parent elements
Paragraph

The following is the XML Schema definition of the **Paragraph.ListLevel** element.

```
<xsd:element name="ListLevel" type="xsd:unsignedInt" minOccurs="0" />
```

#### 2.18.4 Paragraph.ListStyle

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Paragraph.ListStyle** element specifies whether a [Paragraph](#) is part of a **list**, and it identifies the numbering type. This element is optional.

If the **Paragraph.ListStyle** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) and MUST be one of the following:

**None:** Specifies that the paragraph has no list formatting.

**Numbered:** Specifies that the paragraph is part of a numbered list.

**Bulleted:** Specifies that the paragraph is part of a bulleted list.

If the **Paragraph.ListStyle** element is not present, its value is interpreted as "None".

Following is the parent element of the **Paragraph.ListStyle** element.

Parent elements
Paragraph

The following is the XML Schema definition of the **Paragraph.ListStyle** element.

```
<xsd:element name="ListStyle" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None" />
      <xsd:enumeration value="Bulleted" />
      <xsd:enumeration value="Numbered" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

#### 2.18.5 Paragraph.RightIndent

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Paragraph.RightIndent** element specifies the indentation for text within a [Paragraph](#) from the right edge of its containing [Textbox](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**.

Following is the parent element of the **Paragraph.RightIndent** element.

Parent elements
Paragraph

The following is the XML Schema definition of the **Paragraph.RightIndent** element.

```
<xsd:element name="RightIndent" type="xsd:string" minOccurs="0" />
```

### 2.18.6 Paragraph.SpaceAfter

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Paragraph.SpaceAfter** element specifies the vertical spacing after a [Paragraph](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**.

Following is the parent element of the **Paragraph.SpaceAfter** element.

Parent elements
Paragraph

The following is the XML Schema definition of the **Paragraph.SpaceAfter** element.

```
<xsd:element name="SpaceAfter" type="xsd:string" minOccurs="0" />
```

### 2.18.7 Paragraph.SpaceBefore

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Paragraph.SpaceBefore** element specifies the vertical spacing before a [Paragraph](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**.

Following is the parent element of the **Paragraph.SpaceBefore** element.

Parent elements
Paragraph

The following is the XML Schema definition of the **Paragraph.SpaceBefore** element.

```
<xsd:element name="SpaceBefore" type="xsd:string" minOccurs="0" />
```

### 2.18.8 Paragraph.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Paragraph.Style** element specifies style properties for a [Paragraph](#). This element is optional and is of type [Style](#).

Following is the parent element of the **Paragraph.Style** element.

Parent elements
Paragraph

The following is the XML Schema definition of the **Paragraph.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.18.9 Paragraph.TextRuns

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Paragraph.TextRuns** element specifies a collection of [TextRun](#) elements for a [Paragraph](#). This element **MUST** be specified. This element is of type [TextRuns](#).

Following is the parent element of the **Paragraph.TextRuns** element.

Parent elements
Paragraph

The following is the XML Schema definition of the **Paragraph.TextRuns** element.

```
<xsd:element name="TextRuns" type="TextRunsType" minOccurs="1" />
```

### 2.19 TextRuns

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TextRuns** element specifies a collection of [TextRun](#) elements.

The following are the parent and child elements of the **TextRuns** element.

Parent elements
<a href="#">Paragraph</a>

Child elements
<a href="#">TextRuns.TextRun</a>

The following is the XML Schema definition of the **TextRuns** element in RDL 2008/01.

```
<xsd:complexType name="TextRunsType">
  <xsd:sequence>
    <xsd:element name="TextRun" type="TextRunType" minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TextRuns** element in RDL 2010/01 and RDL 2016/01.



```

<xsd:complexType name="TextRunsType">
  <xsd:sequence>
    <xsd:element name="TextRun" type="TextRunType" minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.19.1 TextRuns.TextRun

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TextRuns.TextRun** element is of type [TextRun](#). This element **MUST** be specified at least once within a [TextRuns](#) collection.

Following is the parent element of the **TextRuns.TextRun** element.

Parent elements
<a href="#">TextRuns</a>

The following is the XML Schema definition of the **TextRuns.TextRun** element.

```

<xsd:element name="TextRun" type="TextRunType" minOccurs="1" maxOccurs="unbounded" />

```

### 2.20 TextRun

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TextRun** element specifies the value and formatting of a contiguous span of text. This element **MUST** be specified at least once within a [TextRuns](#) collection.

The following are the parent and child elements of the **TextRun** element.

Parent elements
<a href="#">TextRuns</a>

Child elements
<a href="#">TextRun.ActionInfo</a>
<a href="#">TextRun.Label</a>
<a href="#">TextRun.MarkupType</a>
<a href="#">TextRun.Style</a>
<a href="#">TextRun.ToolTip</a>
<a href="#">TextRun.Value</a>

The following is the XML Schema definition of the **TextRun** element in RDL 2008/01.

```

<xsd:complexType name="TextRunType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">

```

```

<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="Value" type="LocIDStringWithDataAttribute" minOccurs="1" />
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="MarkupType" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **TextRun** element in RDL 2010/01\_and RDL 2016/01.

```

<xsd:complexType name="TextRunType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Value" type="LocIDStringWithDataAttribute" minOccurs="1" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkupType" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.20.1 TextRun.ActionInfo

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TextRun.ActionInfo** element specifies the action for a [TextRun](#). This element is optional. This element is ignored if an [ActionInfo](#) is specified for the parent [Textbox](#), even if the **text box** action resolves to NULL. This element is of type **ActionInfo**.

Following is the parent element of the **TextRun.ActionInfo** element.

Parent elements
TextRun

The following is the XML Schema definition of the **TextRun.ActionInfo** element.

```

<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />

```

### 2.20.2 TextRun.Label

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TextRun.Label** element specifies a label for the placeholder for a [TextRun](#). If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1). The value of this element MUST appear as the display placeholder in the designer tool UI.

Following is the parent element of the **TextRun.Label** element.

Parent elements
TextRun

The following is the XML Schema definition of the **TextRun.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

### 2.20.3 TextRun.MarkupType

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TextRun.MarkupType** element specifies whether markup that appears in [TextRun.Value](#) will be processed. This element is optional. If the **TextRun.MarkupType** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**None:** Specifies that no markup is processed and that all markup is left as is.

**HTML:** Specifies that HTML markup appearing in **TextRun.Value** is processed and displayed in supporting rendering extensions.

If the **TextRun.MarkupType** element is not present, its value is interpreted as "None".

Following is the parent element of the **TextRun.MarkupType** element.

Parent elements
<a href="#">TextRun</a>

The following is the XML Schema definition of the **TextRun.MarkupType** element.

```
<xsd:element name="MarkupType" type="xsd:string" minOccurs="0" />
```

### 2.20.4 TextRun.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TextRun.Style** element specifies style properties for a [TextRun](#). This element is optional. This element is of type [Style](#).

Following is the parent element of the **TextRun.Style** element.

Parent elements
TextRun

The following is the XML Schema definition of the **TextRun.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.20.5 TextRun.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TextRun.ToolTip** element specifies a textual tooltip label for a [TextRun](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **TextRun.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **TextRun.ToolTip** element.

Parent elements
TextRun

The following is the XML Schema definition of the **TextRun.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

## 2.20.6 TextRun.Value

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TextRun.Value** element specifies a value or an expression that is displayed for a [TextRun](#). This element MUST be specified.

The following are the parent element and attributes of the **TextRun.Value** element.

Parent elements
TextRun

Attributes
<a href="#">TextRun.Value.DataType</a>
<a href="#">TextRun.Value.EvaluationMode</a>

The following is the XML Schema definition of the **TextRun.Value** element.

```
<xsd:element name="Value" type="LocIDStringWithDataTypeAttribute" minOccurs="1" />
```

## 2.20.7 TextRun.Value.DataType

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TextRun.Value.DataType** attribute specifies the data type of the [TextRun.Value](#) element if the value of **TextRun.Value** is a constant. This attribute is optional. If this attribute is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1), a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2), a [DateTime](#) ([XMLSCHEMA2] section 3.2.7), an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17), or a [Float](#) ([XMLSCHEMA2] section 3.2.4). If this attribute is not present, its value is interpreted as a **String**.

Following is the parent element of the **TextRun.Value.DataType** attribute.

Parent elements
TextRun.Value

The following is the XML Schema definition of the **TextRun.Value.DataType** attribute.

```
<xsd:attribute name="DataType" use="optional">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Boolean" />
      <xsd:enumeration value="DateTime" />
      <xsd:enumeration value="Integer" />
      <xsd:enumeration value="Float" />
      <xsd:enumeration value="String" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>
```

## 2.20.8 TextRun.Value.EvaluationMode

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TextRun.Value.EvaluationMode** attribute specifies whether [TextRun.Value](#) is treated as an expression or constant value. This attribute is optional. If this attribute is present, its value **MUST** be one of the following:

**Auto:** Specifies that **TextRun.Value** is treated as an expression if it starts with "=" and as a constant otherwise.

**Constant:** Specifies that **TextRun.Value** is treated as a constant value.

If this element is not present, its value is interpreted as "Auto". Following is the parent element of the **TextRun.Value.EvaluationMode** attribute.

Parent elements
TextRun.Value

The following is the XML Schema definition of the **TextRun.Value.EvaluationMode** attribute.

```
<xsd:attribute name="EvaluationMode" type="EvaluationModeType" default="Auto" />
```

## 2.21 ToggleImage

The **ToggleImage** element specifies the initial state (+ or -) of a **toggle image** if the image is displayed as part of a [Textbox](#). If the parent text box is a toggle item for the visibility of another report item, the image **MUST** always be displayed. When the text box or image is clicked, the toggle image **MUST** flip state. This element is optional.

The following are the parent and child elements of the **ToggleImage** element.

Parent elements
Textbox

Child elements
<a href="#">ToggleImage.InitialState</a>

The following is the XML Schema definition of the **ToggleImage** element.

```
<xsd:complexType name="ToggleImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="InitialState" type="xsd:string" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.21.1 ToggleImage.InitialState

The **ToggleImage.InitialState** element specifies the initial state of a [ToggleImage](#). This element **MUST** be specified, and its value **MUST** be a [Boolean](#) ([XMLSCHEMA2/2](#) section 3.2.2) or an expression that evaluates to a **Boolean**.

If **ToggleImage.InitialState** is set to true, the **toggle image** is interpreted as expanded and **MUST** show a minus sign. If **ToggleImage.InitialState** is set to false, the toggle image is interpreted as collapsed and **MUST** show a plus sign.

Following is the parent element of the **ToggleImage.InitialState** element.

Parent elements
ToggleImage

The following is the XML Schema definition of the **ToggleImage.InitialState** element.

```
<xsd:element name="InitialState" type="xsd:string">
```

## 2.22 UserSort

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **UserSort** element specifies an end-user sort control that is displayed as part of a [Textbox](#) in a rendering of a [Report](#). The control allows the user to select a sort direction (ascending, descending, or none).

The following are the parent and child elements of the **UserSort** element.

Parent elements
Textbox

Child elements
<a href="#">UserSort.SortExpression</a>
<a href="#">UserSort.SortExpressionScope</a>
<a href="#">UserSort.SortTarget</a>

The following is the XML Schema definition of the **UserSort** element in RDL 2005/01 and RDL 2008/01.

```

<xsd:complexType name="UserSortType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="SortExpression" type="xsd:string" />
    <xsd:element name="SortExpressionScope" type="xsd:string" minOccurs="0" />
    <xsd:element name="SortTarget" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **UserSort** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="UserSortType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="SortExpression" type="xsd:string" />
    <xsd:element name="SortExpressionScope" type="xsd:string" minOccurs="0" />
    <xsd:element name="SortTarget" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.22.1 UserSort.SortExpression

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **UserSort.SortExpression** element specifies an expression on which to sort. This element **MUST** be specified. Its value **MUST** be a [String](#) ([IXMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **variant**.

The same restrictions that applied to [Filter.FilterExpression](#) instances within a [Group](#) **MUST** also apply to this element. If the expression that this element specifies contains an **aggregate function**, the aggregate function **MUST** have a **scope** that is equal to or that contains the value of [UserSort.SortExpressionScope](#). If the peer **UserSort.SortExpressionScope** element is not defined, the **UserSort.SortExpression** element **MUST NOT** contain aggregate function instances without an explicit scope.

Following is the parent element of the **UserSort.SortExpression** element.

Parent elements
<a href="#">UserSort</a>

The following is the XML Schema definition of the **UserSort.SortExpression** element.

```

<xsd:element name="SortExpression" type="xsd:string" />

```

### 2.22.2 UserSort.SortExpressionScope

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **UserSort.SortExpressionScope** element specifies the name of the scope (**data region** or [Group](#)) in which to evaluate the [UserSort.SortExpression](#). The **UserSort.SortExpressionScope** element is optional.

If the **UserSort.SortExpressionScope** element is specified, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1). If this element is not present, the sort expression will be evaluated and the sort will be performed independently in each **detail scope** within the sort target.

The value of the **UserSort.SortExpressionScope** element MUST be equal to or contained within the current scope. If the containing [Textbox](#) has no current scope (for example, the text box is not contained within any data region or **Group**), the value of this element MUST be equal to or contained within the scope that is defined by the peer [UserSort.SortTarget](#) element.

The value of the **UserSort.SortExpressionScope** element MUST NOT evaluate to a detail scope (for example, a **Group** that has no [Group.GroupExpressions](#) instance). The [DataSet](#) that is associated with **UserSort.SortExpressionScope** MUST be the same as the **DataSet** that is associated with the peer **UserSort.SortTarget** element. Sorting for the [UserSort](#) MUST take place within the **Group** that contains the **UserSort.SortExpressionScope**.

Following is the parent element of the **UserSort.SortExpressionScope** element.

Parent elements
<a href="#">UserSort</a>

The following is the XML Schema definition of the **UserSort.SortExpressionScope** element.

```
<xsd:element name="SortExpressionScope" type="xsd:string" minOccurs="0" />
```

### 2.22.3 UserSort.SortTarget

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **UserSort.SortTarget** element specifies the name of a data region ([Tablix](#), [Chart](#), [GaugePanel](#), [CustomReportItem](#), [List](#), [Table](#), [Matrix](#)), [Group](#), or [DataSet](#) to which to apply the sort. This element is optional.

If the **UserSort.SortTarget** element is specified, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that MUST be the name of the current scope, the name of an ancestor scope, or the name of a peer scope that is a data region. If this element is not present, its value is interpreted as that of the peer scope.

Following is the parent element of the **UserSort.SortTarget** element.

Parent elements
<a href="#">UserSort</a>

The following is the XML Schema definition of the **UserSort.SortTarget** element.

```
<xsd:element name="SortTarget" type="xsd:string" minOccurs="0" />
```

### 2.23 Tablix

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix** element defines a **tablix**, which is a layout grid that consists of columns and rows that can optionally be filtered, sorted, grouped, nested, and repeated.



The tablix incorporates the collective functionality of the [List](#), [Table](#), and [Matrix](#) that were defined in versions of RDL prior to RDL 2008/01.

An element of type **Tablix** MUST NOT be specified within a page header or a page footer.

There are two collections of [TablixMember](#) elements. **TablixMember** elements that have a [TablixRowHierarchy](#) ancestor in the tablix are called tablix **row members**. **TablixMember** elements that have a [TablixColumnHierarchy](#) ancestor in the tablix are called tablix **column members**.

The quantity of [TablixColumn](#) elements within the tablix MUST equal the quantity of tablix column members that do not have a **TablixMember** element descendant. The quantity of **TablixColumn** elements within the tablix MUST equal the quantity of [TablixCell](#) elements within the tablix.

The value of the [Tablix.LayoutDirection](#) element changes the effect that the value of the [Tablix.GroupsBeforeRowHeaders](#) element will have. This is described under **Tablix.GroupsBeforeRowHeaders**.

Columns of tablix **row headers** MUST be laid out before columns that are defined by **TablixColumn** elements, and they MUST be considered in conjunction with the **Tablix.GroupsBeforeRowHeaders** element.

The [Tablix.TablixCorner](#) element MUST be specified if the following conditions are true:

- The tablix has a **TablixRowHierarchy** element child that has a [TablixHeader](#) element descendant.
- The tablix has a **TablixColumnHierarchy** element child that has a **TablixHeader** element descendant.

If either the **TablixRowHierarchy** element or the **TablixColumnHierarchy** element does not have a **TablixHeader** element descendant, the **Tablix** element MUST NOT have a child [TablixCorner](#) element.

If a [TablixCornerRow](#) element is specified, the quantity of descendant [TablixCornerCell](#) elements in the tablix MUST equal the quantity of unique cumulative widths for each tablix row member of the tablix. "Cumulative width" for a tablix row member is defined as the sum of the values of all [TablixHeader.Size](#) elements that are descendants of the tablix row member.

If a **TablixCorner** element is specified, the quantity of descendant **TablixCornerRow** elements in the tablix MUST equal the quantity of unique cumulative heights for each tablix column member of the tablix. "Cumulative height" for a tablix column member is defined as the sum of the values of all **TablixHeader.Size** elements that are descendants of the tablix column member.

Borders and any **background images** that are specified under [Tablix.Style](#) can be repeated on vertical and horizontal page breaks by using the [Tablix.OmitBorderOnPageBreak](#) element.

The informative reference *Understanding Tablix Data Region Areas* [[MSDN-UTDRA](#)] is useful to understand the nomenclature of tablix components.

The following are the parent elements, attribute, and child elements of the **Tablix** element.

Parent elements
<a href="#">ReportItems</a>
<a href="#">CellContents</a>
<a href="#">CustomReportItem.AltReportItem</a>

<b>Attributes</b>
<a href="#">Tablix.Name</a>

<b>Child elements</b>
<a href="#">Tablix.ActionInfo</a>
<a href="#">Tablix.Bookmark</a>
<a href="#">Tablix.CustomProperties</a>
<a href="#">Tablix.DataElementName</a>
<a href="#">Tablix.DataElementOutput</a>
<a href="#">Tablix.DataSetName</a>
<a href="#">Tablix.DocumentMapLabel</a>
<a href="#">Tablix.Filters</a>
<a href="#">Tablix.FixedColumnHeaders</a>
<a href="#">Tablix.FixedRowHeaders</a>
Tablix.GroupsBeforeRowHeaders
<a href="#">Tablix.Height</a>
<a href="#">Tablix.KeepTogether</a>
Tablix.LayoutDirection
<a href="#">Tablix.Left</a>
<a href="#">Tablix.NoRowsMessage</a>
Tablix.OmitBorderOnPageBreak
<a href="#">Tablix.PageBreak</a>
<a href="#">Tablix.PageName</a>
<a href="#">Tablix.RepeatColumnHeaders</a>
<a href="#">Tablix.RepeatRowHeaders</a>
<a href="#">Tablix.RepeatWith</a>
<a href="#">Tablix.SortExpressions</a>
Tablix.Style
<a href="#">Tablix.TablixBody</a>
<a href="#">Tablix.TablixColumnHierarchy</a>
Tablix.TablixCorner
<a href="#">Tablix.TablixRowHierarchy</a>

Child elements
<a href="#">Tablix.ToolTip</a>
<a href="#">Tablix.Top</a>
<a href="#">Tablix.Visibility</a>
<a href="#">Tablix.Width</a>
<a href="#">Tablix.ZIndex</a>

Applies to [RDL 2011/01](#)

Child elements
<a href="#">Tablix.BandLayoutOptions</a>
<a href="#">Tablix.BottomMargin</a>
<a href="#">Tablix.CanScroll</a>
<a href="#">Tablix.LeftMargin</a>
<a href="#">Tablix.Relationship</a>
<a href="#">Tablix.ReportSlicerState</a>
<a href="#">Tablix.RightMargin</a>
<a href="#">Tablix.TopMargin</a>

The following is the XML Schema definition of the **Tablix** element in RDL 2008/01.

```
<xsd:complexType name="TablixType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixCorner" type="TablixCornerType" minOccurs="0" />
    <xsd:element name="TablixBody" type="TablixBodyType" minOccurs="0" />
    <xsd:element name="TablixColumnHierarchy" type="TablixHierarchyType" minOccurs="1" />
    <xsd:element name="TablixRowHierarchy" type="TablixHierarchyType" minOccurs="1" />
    <xsd:element name="LayoutDirection" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="LTR" />
          <xsd:enumeration value="RTL" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GroupsBeforeRowHeaders" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="RepeatColumnHeaders" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="RepeatRowHeaders" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="FixedColumnHeaders" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="FixedRowHeaders" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

```

<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
<xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **Tablix** element in RDL 2010/01 and RDL 2016/01.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```

<xsd:complexType name="TablixType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixCorner" type="TablixCornerType" minOccurs="0" />
    <xsd:element name="TablixBody" type="TablixBodyType" minOccurs="0" />
    <xsd:element name="TablixColumnHierarchy" type="TablixHierarchyType" minOccurs="1" />
    <xsd:element name="TablixRowHierarchy" type="TablixHierarchyType" minOccurs="1" />
    <xsd:element name="LayoutDirection" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="LTR" />
          <xsd:enumeration value="RTL" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GroupsBeforeRowHeaders" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="RepeatColumnHeaders" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="RepeatRowHeaders" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="FixedColumnHeaders" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="FixedRowHeaders" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>

```

```

<xsd:element name="PageName" type="xsd:string" minOccurs="0" />
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
<xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.23.1 Tablix.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.Name** attribute specifies a unique identifier for a [Tablix](#). This attribute **MUST** be specified and its value **MUST** be a **NormalizedString** that is a case-sensitive **CLS-compliant identifier [UTR15]** that is unique among report items and **scope** names.

Following is the parent element of the **Tablix.Name** attribute.

Parent elements
<a href="#">Tablix</a>

The following is the XML Schema definition of the **Tablix.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.23.2 Tablix.ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.ActionInfo** element is ignored.

Following is the parent element of the **Tablix.ActionInfo** element.

Parent elements
<a href="#">Tablix</a>

The following is the XML Schema definition of the **Tablix.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.23.3 Tablix.BandLayoutOptions

*Applies to [RDL 2011/01](#)*

The **Tablix.BandLayoutOptions** element specifies layout options for a [Tablix](#) when it is displayed as a **band**. The **Tablix.BandLayoutOptions** element is optional and MUST NOT be specified more than once.

If this element is specified, it is of type [BandLayoutOptions](#).

Following is the parent element of the **Tablix.BandLayoutOptions** element.

Parent elements
Tablix

### 2.23.4 Tablix.BottomMargin

*Applies to [RDL 2011/01](#)*

The **Tablix.BottomMargin** element specifies the width of the [Tablix](#)'s bottom margin when it is displayed as a **band**. The **Tablix.BottomMargin** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [RdlSize](#). If this element is not specified, its value is interpreted as the minimum size.

If [Tablix.BandLayoutOptions](#) is not specified, the **Tablix.BottomMargin** element is ignored.

Following is the parent element of the **Tablix.BottomMargin** element.

Parent elements
Tablix

### 2.23.5 Tablix.Bookmark

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.Bookmark** element specifies a bookmark for a [Tablix](#). This element is optional and MUST NOT be specified more than once. If this element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String** or to null.

Following is the parent element of the **Tablix.Bookmark** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.Bookmark** element.

```
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
```

## 2.23.6 Tablix.CanScroll

*Applies to [RDL 2011/01](#)*

The **Tablix.CanScroll** element specifies whether a [Tablix](#) that has a scrollbar. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Tablix.CanScroll** element.

Parent elements
<a href="#">Tablix</a>

The following is the XML Schema definition of the **Tablix.CanScroll** element.

```
<xsd:element name="CanScroll" type="xsd:boolean" minOccurs="0" />
```

## 2.23.7 Tablix.CustomProperties

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.CustomProperties** element specifies extended information in a collection of name/value pairs. This element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [CustomProperties](#).

Following is the parent element of the **Tablix.CustomProperties** element.

Parent elements
<a href="#">Tablix</a>

The following is the XML Schema definition of the **Tablix.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

## 2.23.8 Tablix.DataElementName

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.DataElementName** element specifies a name to use for a **data element** of a [Tablix](#) in a **data rendering**. This element is optional and MUST NOT be specified more than once.

If the **Tablix.DataElementName** element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#) that is unique within the containing object of the **tablix**. If this element is not specified, its value is interpreted as the value of the **Name** attribute of the **tablix**.

Following is the parent element of the **Tablix.DataElementName** element.

Parent elements
<a href="#">Tablix</a>

The following is the XML Schema definition of the **Tablix.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.23.9 Tablix.DataElementOutput

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.DataElementOutput** element specifies whether a [Tablix](#) is included in a **data rendering**. This element is optional and MUST NOT be specified more than once. If this element is specified, its value MUST be of type [String](#) ([XMLSCHEMA2/2] section 3.2.1) and MUST be one of the following values:

**Output:** The tablix data appears in the data rendering output.

**NoOutput:** The tablix data does not appear in the data rendering output.

**Auto** (default): If the value of the grandchild [Visibility.Hidden](#) element of the **Tablix** element is the **String** literal value of "true" (not an expression) and if a grandchild [Visibility.ToggleItem](#) element is not specified for the **Tablix**, the value of **Tablix.DataElementOutput** value is interpreted as "NoOutput".

If the value of the grandchild **Visibility.Hidden** child element of the **Tablix** element is set to the **String** literal value of "true" (not an expression) and if the tablix is in a non-toggleable **member** of another tablix, the value of the **Tablix.DataElementOutput** is interpreted as "NoOutput".

For all other cases, the value of **Tablix.DataElementOutput** is interpreted as "Output".

If the **Tablix.DataElementOutput** element is not specified, its value is interpreted as "Auto".

Following is the parent element of the **Tablix.DataElementOutput** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.23.10 Tablix.DataSetName

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.DataSetName** element specifies the name of the [DataSet](#) to use for a [Tablix](#). The **Tablix.DataSetName** element is optional and MUST NOT be specified more than once. If this element is specified, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1), and the value of this element MUST be one of the [DataSet.Name](#) attribute values of the tablix.

If the **Tablix** has an ancestor, the value of the **Tablix.DataSetName** element is interpreted as the **DataSet.Name** for the containing scope (**DataRegion**, [Group](#), or **Cell**).<7>



If the tablix has no tablix ancestor and if there is more than one **DataSet** for the containing [Report](#), the value of the **Tablix.DataSetName** element MUST be specified.

If no value is specified for the **Tablix.DataSetName** element and if the containing report has only one **DataSet** specified, the value of **Tablix.DataSetName** is interpreted as being the value of that **DataSet.Name** attribute.

Following is the parent element of the **Tablix.DataSetName** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
```

### 2.23.11 Tablix.DocumentMapLabel

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.DocumentMapLabel** element specifies a **document map** label to identify a [Tablix](#) within the rendered [Report](#). This element is optional and MUST NOT be specified more than once. If this element is specified, it MUST be of type [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Following is the parent element of the **Tablix.DocumentMapLabel** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.DocumentMapLabel** element.

```
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
```

### 2.23.12 Tablix.Filters

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.Filters** element specifies a collection of **filter** expressions to apply to the data for each row of a [Tablix](#). This element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [Filters](#).

Following is the parent element of the **Tablix.Filters** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

### 2.23.13 **Tablix.FixedColumnHeaders**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.FixedColumnHeaders** element specifies whether tablix **column headers** remain visible when a [Tablix](#) is partially scrolled off a [Page](#). This element is optional and MUST NOT be specified more than once.

If the **Tablix.FixedColumnHeaders** element is specified, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). A value of true specifies that the tablix column headers remain visible. If this element is not specified, its value is interpreted as false.

If the [Tablix.BandLayoutOptions](#) element is specified, the **Tablix.FixedColumnHeaders** element is ignored.

Following is the parent element of the **Tablix.FixedColumnHeaders** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.FixedColumnHeaders** element.

```
<xsd:element name="FixedColumnHeaders" type="xsd:boolean" minOccurs="0" />
```

### 2.23.14 **Tablix.FixedRowHeaders**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.FixedRowHeaders** element specifies whether [Tablix](#) row headers remain visible when a tablix is partially scrolled off a [Page](#). This element is optional and MUST NOT be specified more than once. If this element is specified, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). A value of true specifies that the tablix row headers remain visible. If this element is not specified, its value is interpreted as false.

If the [Tablix.BandLayoutOptions](#) element is specified, the **Tablix.FixedRowHeaders** element is ignored.

Following is the parent element of the **Tablix.FixedRowHeaders** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.FixedRowHeaders** element.

```
<xsd:element name="FixedRowHeaders" type="xsd:boolean" minOccurs="0" />
```

### 2.23.15 **Tablix.GroupsBeforeRowHeaders**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.GroupsBeforeRowHeaders** element specifies the count of **dynamic member** columns of the first [Tablix](#) column group to render before the tablix row header. This element is optional and MUST NOT be specified more than once.

If the **Tablix.GroupsBeforeRowHeaders** element is specified, its value MUST be an **UnsignedInt**. If this element is specified, its value MUST be greater than or equal to 0 and less than or equal to 2147483647.

If the [Tablix.LayoutDirection](#) element value is "LTR", the **Tablix.GroupsBeforeRowHeaders** element specifies the number of instances of the first tablix **column member** to appear to the left of the tablix row headers.

If the **Tablix.LayoutDirection** element value is "RTL", the **Tablix.GroupsBeforeRowHeaders** element specifies the number of instances of the first tablix column member to appear to the right of the tablix row headers.

If the first tablix column member is a **static member**, the value of the **Tablix.GroupsBeforeRowHeaders** element is interpreted as 0. If the first tablix column member has a child [TablixMember.FixedData](#) element whose value is true, the value of the **Tablix.GroupsBeforeRowHeaders** element MUST be 0.

If the **Tablix.GroupsBeforeRowHeaders** element is not specified, its value is interpreted as 0.

If the [Tablix.BandLayoutOptions](#) element is specified, the **Tablix.GroupsBeforeRowHeaders** element is ignored.

Following is the parent element of the **Tablix.GroupsBeforeRowHeaders** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.GroupsBeforeRowHeaders** element.

```
<xsd:element name="GroupsBeforeRowHeaders" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.23.16 Tablix.Height

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.Height** element specifies the height of a [Tablix](#). This element is optional and MUST NOT be specified more than once.

If the **Tablix.Height** element is specified, it MUST be of type [RdlSize](#). If this element is not specified, the value of this element is derived from the following:

- The first-degree descendants of the [TablixRow.Height](#) element.
- The [TablixHeader.Size](#) element first-degree descendants of the [TablixColumnHierarchy](#) element.

Following is the parent element of the **Tablix.Height** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="0" />
```

### 2.23.17 **Tablix.KeepTogether**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.KeepTogether** element specifies whether the renderer attempts to render the entire [Tablix](#) on one [Page](#). This element is optional and MUST NOT be specified more than once.

If the **Tablix.KeepTogether** element is specified, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). A value of true specifies that the renderer attempts to render the entire tablix on one page. If this element is not specified, its value is interpreted as false.

Following is the parent element of the **Tablix.KeepTogether** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.KeepTogether** element.

```
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
```

### 2.23.18 **Tablix.LayoutDirection**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.LayoutDirection** element specifies the layout direction of [Tablix](#) columns. This element is optional and MUST NOT be specified more than once. If this element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1), which MUST be one of the following:

**LTR:** Tablix columns are laid out in sequence, starting at the left and progressing to the right. [<8>](#)

**RTL:** Tablix columns are laid out in sequence, starting at the right and progressing to the left.

If the **Tablix.LayoutDirection** element is not specified, its value is interpreted as "LTR" regardless of whether the implementation is executed on a left-to-right or right-to-left operating system.

**Note** The value of the **Tablix.LayoutDirection** element changes the effect that the value of the [Tablix.GroupsBeforeRowHeaders](#) element has. This is described under **Tablix.GroupsBeforeRowHeaders**.

Following is the parent element of the **Tablix.LayoutDirection** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.LayoutDirection** element.

```
<xsd:element name="LayoutDirection" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="LTR" />
      <xsd:enumeration value="RTL" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.23.19 **Tablix.Left**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.Left** element specifies the distance of the [Tablix](#) from the left of the containing object. This element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [RdlSize](#). If this element is not specified, its value is interpreted as the minimum size.

Following is the parent element of the **Tablix.Left** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.Left** element.

```
<xsd:element name="Left" type="SizeType" minOccurs="0" />
```

### 2.23.20 **Tablix.LeftMargin**

*Applies to [RDL 2011/01](#)*

The **Tablix.LeftMargin** element specifies the width of the [Tablix](#)'s left margin when it is displayed as a **band**. The **Tablix.LeftMargin** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [RdlSize](#). If this element is not specified, its value is interpreted as the minimum size.

If [Tablix.BandLayoutOptions](#) is not specified, the **Tablix.LeftMargin** element is ignored.

Following is the parent element of the **Tablix.LeftMargin** element.

Parent elements
Tablix

### 2.23.21 **Tablix.NoRowsMessage**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.NoRowsMessage** element specifies text to render instead of the [Tablix](#) layout when no data is available. This element is optional and MUST NOT be specified more than once. If this element is specified, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String** or to null.

Following is the parent element of the **Tablix.NoRowsMessage** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.NoRowsMessage** element.

```
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
```

### 2.23.22 **Tablix.OmitBorderOnPageBreak**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.OmitBorderOnPageBreak** element specifies how borders and **background images** that are specified under the [Tablix.Style](#) element are rendered when a rendered [Tablix](#) spans multiple pages of the output format. The **Tablix.OmitBorderOnPageBreak** element is optional and MUST NOT be specified more than once.

If the **Tablix.Style** element is not specified, the **Tablix.OmitBorderOnPageBreak** element is ignored. If the **Tablix.OmitBorderOnPageBreak** element is specified, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2).

If borders are specified to be rendered by the **Tablix.Style** element, a value of false specifies that borders are to be rendered along any page breaks, and a value of true specifies that borders are not to be rendered along any page breaks.

If a repeating [BackgroundImage](#) is specified to be rendered by the **Tablix.Style** element, a value of false specifies that the background image is restarted on each page of the tablix, and a value of true specifies that the background image continues as if it had been sliced at the page break.

If the **Tablix.OmitBorderOnPageBreak** element is not specified, its value is interpreted as false.

Following is the parent element of the **Tablix.OmitBorderOnPageBreak** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.OmitBorderOnPageBreak** element.

```
<xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
```

### 2.23.23 **Tablix.PageBreak**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.PageBreak** element specifies the [PageBreak](#) behavior of a [Tablix](#). This element is optional and MUST NOT be specified more than once. If this element is specified, it is of type **PageBreak**.

Following is the parent element of the **Tablix.PageBreak** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.PageBreak** element.

```
<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
```

### 2.23.24 **Tablix.PageName**

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **Tablix.PageName** element specifies the value to use for the name of a paginated page. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Tablix.PageName** element.

Parent elements
<a href="#">Tablix</a>

The following is the XML Schema definition of the **Tablix.PageName** element.

```
<xsd:element name="PageName" type="xsd:string" minOccurs="0" />
```

### 2.23.25 **Tablix.Relationship**

*Applies to [RDL 2011/01](#)*

The **Tablix.Relationship** element specifies a **relationship** to use for correlating data in a [Tablix](#) with the data in the containing scope. The **Tablix.Relationship** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [Relationship](#). This element is ignored if the dataset for this **Tablix** is the same as the dataset for each containing scope. The **Tablix.Relationship** element MUST NOT be specified if there is no containing scope.

Following is the parent element of the **Tablix.Relationship** element.

Parent elements
<a href="#">Tablix</a>

The following is the XML Schema definition of the **Tablix.Relationship** element

```
<xsd:element name="Relationship" type="RelationshipType" minOccurs="0" />
```

### 2.23.26 **Tablix.RepeatColumnHeaders**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.RepeatColumnHeaders** element specifies whether tablix **column headers** are repeated on each page where the column traverses a page break. This element is optional and MUST NOT be specified more than once.

If the **Tablix.RepeatColumnHeaders** element is specified, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). A value of true specifies that the tablix column headers are to be repeated. If this element is not specified, its value is interpreted as false.

If the [Tablix.BandLayoutOptions](#) element is specified, the **Tablix.RepeatColumnHeaders** element is ignored.

Following is the parent element of the **Tablix.RepeatColumnHeaders** element.

Parent elements
<a href="#">Tablix</a>

The following is the XML Schema definition of the **Tablix.RepeatColumnHeaders** element.

```
<xsd:element name="RepeatColumnHeaders" type="xsd:boolean" minOccurs="0" />
```

### 2.23.27 **Tablix.RepeatRowHeaders**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.RepeatRowHeaders** element specifies whether [Tablix](#) row headers are repeated on each page where the row traverses a page break. This element is optional and MUST NOT be specified more than once.

If the **Tablix.RepeatRowHeaders** element is specified, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). A value of true specifies that the tablix row headers are to be repeated. If this element is not specified, its value is interpreted as false.

If the [Tablix.BandLayoutOptions](#) element is specified, the **Tablix.RepeatRowHeaders** element is ignored.

Following is the parent element of the **Tablix.RepeatRowHeaders** element.

Parent elements
<a href="#">Tablix</a>

The following is the XML Schema definition of the **Tablix.RepeatRowHeaders** element.

```
<xsd:element name="RepeatRowHeaders" type="xsd:boolean" minOccurs="0" />
```

### 2.23.28 **Tablix.RepeatWith**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.RepeatWith** element MUST NOT be specified.

Following is the parent element of the **Tablix.RepeatWith** element.

Parent elements
<a href="#">Tablix</a>

The following is the XML Schema definition of the **Tablix.RepeatWith** element.

```
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
```

### 2.23.29 **Tablix.ReportSlicerState**

**Applies to** [RDL 2011/01](#)

The **Tablix.ReportSlicerState** element specifies the filter state for a [Tablix](#) that filters data for other report items. The **Tablix.ReportSlicerState** element is optional and MUST NOT be specified more than once. This element is defined in [\[MS-RDLRS\]](#) section 2.31.

### 2.23.30 **Tablix.RightMargin**

**Applies to** [RDL 2011/01](#)



The **Tablix.RightMargin** element specifies the width of the [Tablix](#)'s right margin when it is displayed as a **band**. The **Tablix.RightMargin** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [RdlSize](#). If this element is not specified, its value is interpreted as the minimum size.

If [Tablix.BandLayoutOptions](#) is not specified, the **Tablix.RightMargin** element is ignored.

Following is the parent element of the **Tablix.RightMargin** element.

Parent elements
Tablix

### 2.23.31 **Tablix.SortExpressions**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.SortExpressions** element specifies a collection of expressions that are applied to the filtered data of a [Tablix](#) to order the data. This element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [SortExpressions](#).

Following is the parent element of the **Tablix.SortExpressions** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.SortExpressions** element.

```
<xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
```

### 2.23.32 **Tablix.Style**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.Style** element specifies the [Style](#) element of a [Tablix](#). The **Tablix.Style** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type **Style**.

See also [Tablix.OmitBorderOnPageBreak](#).

Following is the parent element of the **Tablix.Style** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.23.33 **Tablix.TablixBody**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.TablixBody** element specifies the body of a [Tablix](#). The **Tablix.TablixBody** element MUST be specified if the **Tablix** has both a [TablixRowHierarchy](#) element child with a [TablixHeader](#) element descendant and a [TablixColumnHierarchy](#) element child with a **TablixHeader** element descendant. If the **Tablix.TablixBody** element is specified, it is of type [TablixBody](#).

Following is the parent element of the **Tablix.TablixBody** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.TablixBody** element.

```
<xsd:element name="TablixBody" type="TablixBodyType" minOccurs="0" />
```

### 2.23.34 **Tablix.TablixColumnHierarchy**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.TablixColumnHierarchy** element specifies the column hierarchy of a [Tablix](#). This element MUST be specified exactly once and is of type [TablixHierarchy](#).

Following is the parent element of the **Tablix.TablixColumnHierarchy** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.TablixColumnHierarchy** element.

```
<xsd:element name="TablixColumnHierarchy" type="TablixHierarchyType" minOccurs="1" />
```

### 2.23.35 **Tablix.TablixCorner**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.TablixCorner** element specifies the [TablixCorner](#) of a [Tablix](#). The **Tablix.TablixCorner** element is optional and MUST NOT be specified more than once.

The **Tablix.TablixCorner** element MUST be specified if the tablix has both a [TablixRowHierarchy](#) element child with a [TablixHeader](#) element descendant, and a [TablixColumnHierarchy](#) element child with a **TablixHeader** element descendant. If this element is specified, it is of type **TablixCorner**.

Following is the parent element of the **Tablix.TablixCorner** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.TablixCorner** element.

```
<xsd:element name="TablixCorner" type="TablixCornerType" minOccurs="0" />
```

### 2.23.36 **Tablix.TablixRowHierarchy**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.TablixRowHierarchy** element specifies the row hierarchy of a [Tablix](#). This element MUST be specified exactly once and is of type [TablixHierarchy](#).

Following is the parent element of the **Tablix.TablixRowHierarchy** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.TablixRowHierarchy** element.

```
<xsd:element name="TablixRowHierarchy" type="TablixHierarchyType" minOccurs="1" />
```

### 2.23.37 **Tablix.ToolTip**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.ToolTip** element specifies a tooltip for a [Tablix](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **Tablix.ToolTip** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Following is the parent element of the **Tablix.ToolTip** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

### 2.23.38 **Tablix.Top**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Tablix.Top** element specifies the distance of a [Tablix](#) from the top of the containing object. This element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [RdlSize](#). If this element is not specified, its value is interpreted as the minimum size.

Following is the parent element of the **Tablix.Top** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.Top** element.

```
<xsd:element name="Top" type="SizeType" minOccurs="0" />
```

### 2.23.39 **Tablix.TopMargin**

**Applies to** [RDL 2011/01](#)

The **Tablix.TopMargin** element specifies the width of the [Tablix](#)'s top margin when it is displayed as a band. The **Tablix.TopMargin** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [RdlSize](#). If this element is not specified, its value is interpreted as the minimum size.

If [Tablix.BandLayoutOptions](#) is not specified, the **Tablix.TopMargin** element is ignored.

Following is the parent element of the **Tablix.TopMargin** element.

Parent elements
Tablix

### 2.23.40 **Tablix.Visibility**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.Visibility** element specifies the presentational presence of a [Tablix](#). This element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [Visibility](#).

Following is the parent element of the **Tablix.Visibility** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

### 2.23.41 **Tablix.Width**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.Width** element specifies the width of a [Tablix](#). This element is optional and MUST NOT be specified more than once.

If the **Tablix.Width** element is specified, it is of type [RdlSize](#). If this element is not specified, the value of this element is derived from the following items:

- The [TablixColumn.Width](#) element first-degree descendants of the tablix.
- The [TablixHeader.Size](#) element first-degree descendants of the [TablixRowHierarchy](#) element of the tablix.

Following is the parent element of the **Tablix.Width** element.

Parent elements
Tablix

The following is the XML Schema definition of the **Tablix.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="0" />
```

### 2.23.42 Tablix.ZIndex

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Tablix.ZIndex** element specifies the **ZIndex** of a [Tablix](#). This element is optional and MUST NOT be specified more than once. If this element is specified, its value MUST be an **UnsignedInt**.

Following is the parent element of the **Tablix.ZIndex** element.

Parent elements
<a href="#">Tablix</a>

The following is the XML Schema definition of the **Tablix.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.24 TablixBody

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixBody** element defines the rows and columns of a [Tablix](#).

**Note** The grid of a rendered **tablix** can have other rows or columns defined by [TablixHeader](#) elements.

The following are the parent and child elements of the **TablixBody** element.

Parent elements
<a href="#">Tablix</a>

Child elements
<a href="#">TablixBody.TablixColumns</a>
<a href="#">TablixBody.TablixRows</a>

The following is the XML Schema definition of the **TablixBody** element in RDL 2008/01.

```
<xsd:complexType name="TablixBodyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixColumns" type="TablixColumnsType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="TablixRows" type="TablixRowsType" minOccurs="1" maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TablixBody** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="TablixBodyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixColumns" type="TablixColumnsType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="TablixRows" type="TablixRowsType" minOccurs="1" maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.24.1 TablixBody.TablixColumns

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixBody.TablixColumns** element specifies the column of the body of a [Tablix](#). This element MUST be specified exactly once and is of type [TablixColumns](#).

Following is the parent element of the **TablixBody.TablixColumns** element.

Parent elements
<a href="#">TablixBody</a>

The following is the XML Schema definition of the **TablixBody.TablixColumns** element.

```
<xsd:element name="TablixColumns" type="TablixColumnsType" minOccurs="1" maxOccurs="1" />
```

### 2.24.2 TablixBody.TablixRows

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixBody.TablixRows** element specifies the rows of the body of a [Tablix](#). This element MUST be specified exactly once and is of type [TablixRows](#).

Following is the parent element of the **TablixBody.TablixRows** element.

Parent elements
<a href="#">TablixBody</a>

The following is the XML Schema definition of the **TablixBody.TablixRows** element.

```
<xsd:element name="TablixRows" type="TablixRowsType" minOccurs="1" maxOccurs="1" />
```

## 2.25 TablixColumns

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixColumns** parent element defines the collection of columns of the body of a [Tablix](#).

The following are the parent and child elements of the **TablixColumns** element.

<b>Parent elements</b>
------------------------

<a href="#">TablixBody</a>
----------------------------

<b>Child elements</b>
-----------------------

<a href="#">TablixColumns.TablixColumn</a>
--

The following is the XML Schema definition of the **TablixColumns** element in RDL 2008/01.

```
<xsd:complexType name="TablixColumnsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="TablixColumn" type="TablixColumnType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TablixColumns** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="TablixColumnsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="TablixColumn" type="TablixColumnType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.25.1 TablixColumns.TablixColumn

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixColumns.TablixColumn** element specifies the columns of a [Tablix](#). This element MUST be specified at least once. This element is of type [TablixColumn](#). The quantity of **TablixColumn** elements MUST equal the quantity of tablix **column members** that do not have a [TablixMember](#) element descendant.

Following is the parent element of the **TablixColumns.TablixColumn** element.

<b>Parent elements</b>
------------------------

<a href="#">TablixColumns</a>
-------------------------------

The following is the XML Schema definition of the **TablixColumns.TablixColumn** element.

```
<xsd:element name="TablixColumn" type="TablixColumnType" minOccurs="1"
  maxOccurs="unbounded" />
```

### 2.26 TablixColumn

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixColumn** element defines a column in the body section of a [Tablix](#).

The following are the parent and child elements of the **TablixColumn** element.

Parent elements
<a href="#">TablixColumns</a>

Child elements
<a href="#">TablixColumn.Width</a>

The following is the XML Schema definition of the **TablixColumn** element in RDL 2008/01

```
<xsd:complexType name="TablixColumnType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Width" type="SizeType" minOccurs="1" maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TablixColumn** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="TablixColumnType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Width" type="SizeType" minOccurs="1" maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.26.1 TablixColumn.Width

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixColumn.Width** element specifies the width of a column in a [Tablix](#). This element MUST be specified exactly once and is of type [RdlSize](#).

Following is the parent element of the **TablixColumn.Width** element.

Parent elements
<a href="#">TablixColumn</a>

The following is the XML Schema definition of the [TablixColumns](#) element.

```
<xsd:element name="Width" type="SizeType" minOccurs="1" maxOccurs="1" />
```

### 2.27 TablixRows

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)



The **TablixRows** parent element defines the rows of the tablix body as a collection of [TablixRow](#) elements.

The following are the parent and child elements of the **TablixRows** element.

Parent elements
<a href="#">TablixBody</a>

Child elements
<a href="#">TablixRows.TablixRow</a>

The following is the XML Schema definition of the **TablixRows** element in RDL 2008/01

```
<xsd:complexType name="TablixRowsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="TablixRow" type="TablixRowType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TablixRows** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="TablixRowsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="TablixRow" type="TablixRowType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.27.1 TablixRows.TablixRow

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixRows.TablixRow** element specifies a row in the [TablixBody](#). This element MUST be specified at least once. This element is of type [TablixRow](#).

The quantity of **TablixRow** elements MUST equal the quantity of [TablixMember](#) elements in the [TablixRowHierarchy](#) that do not have a **TablixMember** element descendant.

Following is the parent element of the **TablixRows.TablixRow** element.

Parent elements
<a href="#">TablixRows</a>

The following is the XML Schema definition of the **TablixRows.TablixRow** element.

```
<xsd:element name="TablixRow" type="TablixRowType" minOccurs="1"
```

```
maxOccurs="unbounded" />
```

## 2.28 TablixRow

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixRow** element defines a collection of [TablixCell](#) elements that form the [TablixBody](#).

The following are the parent and child elements of the **TablixRow** element.

Parent elements
<a href="#">TablixRows</a>

Child elements
<a href="#">TablixRow.Height</a>
<a href="#">TablixRow.TablixCells</a>

The following is the XML Schema definition of the **TablixRow** element in RDL 2008/01.

```
<xsd:complexType name="TablixRowType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Height" type="SizeType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="TablixCells" type="TablixCellsType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TablixRow** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="TablixRowType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Height" type="SizeType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="TablixCells" type="TablixCellsType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.28.1 TablixRow.Height

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixRow.Height** element specifies the height of a row in a [Tablix](#). This element MUST be specified exactly once and is of type [RdlSize](#).

Following is the parent element of the **TablixRow.Height** element.

<b>Parent elements</b>
------------------------

<a href="#">TablixRow</a>
---------------------------

The following is the XML Schema definition of the **TablixRow.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="1" maxOccurs="1" />
```

## 2.28.2 TablixRow.TablixCells

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixRow.TablixCells** element specifies a collection of [TablixCell](#) elements of a row in the body of a [Tablix](#). The **TablixRow.TablixCells** element MUST be specified exactly once and is of type [TablixCells](#).

Following is the parent element of the **TablixRow.TablixCells** element.

<b>Parent elements</b>
------------------------

<a href="#">TablixRow</a>
---------------------------

The following is the XML Schema definition of the **TablixRow.TablixCells** element.

```
<xsd:element name="TablixCells" type="TablixCellsType" minOccurs="1"
maxOccurs="1" />
```

## 2.29 TablixCells

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixCells** parent element defines the collection of cells in a row of the body section of a [Tablix](#).

The following are the parent and child elements of the **TablixCells** element.

<b>Parent elements</b>
------------------------

<a href="#">TablixRow</a>
---------------------------

<b>Child elements</b>
-----------------------

<a href="#">TablixCells.TablixCell</a>
--

The following is the XML Schema definition of the **TablixCells** element in RDL 2008/01.

```
<xsd:complexType name="TablixCellsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="TablixCell" type="TablixCellType" minOccurs="1"
maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TablixCells** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="TablixCellsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="TablixCell" type="TablixCellType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.29.1 TablixCells.TablixCell

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixCells.TablixCell** element specifies a cell of a [Tablix](#). This element MUST be specified at least once. The value of this element MUST be of type [TablixCell](#). The quantity of **TablixCell** elements MUST equal the quantity of tablix **column members** that do not have a [TablixMember](#) element descendant.

Following is the parent element of the **TablixCells.TablixCell** element.

Parent elements
<a href="#">TablixCells</a>

The following is the XML Schema definition of the **TablixCells.TablixCell** element.

```
<xsd:element name="TablixCell" type="TablixCellType" minOccurs="1" maxOccurs="unbounded" />
```

### 2.30 TablixCell

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixCell** parent element defines the content of a cell in the [TablixBody](#).

The following are the parent and child elements of the **TablixCell** element.

Parent elements
<a href="#">TablixCells</a>

Child elements
<a href="#">TablixCell.CellContents</a>
<a href="#">TablixCell.DataElementName</a>
<a href="#">TablixCell.DataElementOutput</a>

**Applies to** [RDL 2011/01](#)

<b>Child elements</b>
<a href="#">TablixCell.DataSetName</a>
<a href="#">TablixCell.Relationships</a>

The following is the XML Schema definition of the **TablixCell** element in RDL 2008/01.

```
<xsd:complexType name="TablixCellType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="CellContents" type="CellContentsType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" maxOccurs="1" />
    <xsd:element name="DataElementOutput" minOccurs="0" maxOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TablixCell** element in RDL 2010/01 and RDL 2016/01.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:complexType name="TablixCellType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="CellContents" type="CellContentsType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" maxOccurs="1" />
    <xsd:element name="DataElementOutput" minOccurs="0" maxOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.30.1 TablixCell.CellContents

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixCell.CellContents** element specifies the [CellContents](#) of a [Tablix](#). This element is of type **CellContents**.

The value of a [CellContents.ColSpan](#) child element, decremented by 1, specifies the quantity of sequential sibling [TablixCell](#) elements that MUST NOT have a child **TablixCell.CellContents** element specified. The **TablixCell.CellContents** element MUST be specified exactly once, unless prohibited by a **CellContents.ColSpan** value.

Following is the parent element of the **TablixCell.CellContents** element.

Parent elements
TablixCell

The following is the XML Schema definition of the **TablixCell.CellContents** element.

```
<xsd:element name="CellContents" type="CellContentsType" minOccurs="0"
maxOccurs="1" />
```

### 2.30.2 TablixCell.DataElementName

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TablixCell.DataElementName** element specifies the name to use for the **data element** of the [TablixCell](#) in the output of a **data rendering**. This element is optional and MUST NOT be specified more than once.

If the **TablixCell.DataElementName** element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that SHOULD<9> be a **CLS-compliant identifier** [\[UTR15\]](#) that is unique within the same associated [TablixMember](#) element.

The value of this element is ignored if the parent **TablixCell** element does not have a child [CellContents](#) element specified. If the **TablixCell.DataElementName** element is not specified, its value is interpreted as "Cell".

Following is the parent element of the **TablixCell.DataElementName** element.

Parent elements
TablixCell

The following is the XML Schema definition of the **TablixCell.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"
maxOccurs="1" />
```

### 2.30.3 TablixCell.DataElementOutput

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TablixCell.DataElementOutput** element specifies the inclusion of the [TablixCell](#) in a **data rendering**. This element is optional and MUST NOT be specified more than once.

If this element is specified, its value MUST be one of the following:

**ContentsOnly** (default): The **TablixCell** does not appear in the data rendering output, but its contents appear in the data rendering output as if they were in the container of the **TablixCell** element.

**Output:** The **TablixCell** appears in the data rendering output.

**NoOutput:** The **TablixCell** does not appear in the data rendering output, and its contents do not appear in the data rendering output.

If this element is not specified, its value is interpreted as "ContentsOnly".

Following is the parent element of the **TablixCell.DataElementOutput** element.

Parent elements
TablixCell

The following is the XML Schema definition of the **TablixCell.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0" maxOccurs="1">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.30.4 TablixCell.DataSetName

**Applies to** [RDL 2011/01](#)

The **TablixCell.DataSetName** element specifies which [DataSet](#) to use for the [TablixCell](#). The **TablixCell.DataSetName** element is optional and MUST NOT be specified more than once.

If the **TablixCell.DataSetName** element is specified, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is a **CLS-compliant identifier** [UTR15]. If this element is not present, its value is interpreted as the **DataSetName** of the containing scope. The **TablixCell.DataSetName** element MUST be specified unless all containing scopes have the same **DataSet**. The **TablixCell.DataSetName** element is ignored if the **DataSet** for this **TablixCell** is the same as the **DataSet** for all containing scopes. <10><11>

Following is the parent element of the **TablixCell.DataSetName** element.

Parent elements
TablixCell

The following is the XML Schema definition of the **TablixCell.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
```

### 2.30.5 TablixCell.Relationships

**Applies to** [RDL 2011/01](#)

The **TablixCell.Relationships** element specifies a collection of the **relationships** to use for correlating data in this [TablixCell](#) with the data in the containing scopes. The **TablixCell.Relationships** element is optional and MUST NOT be specified more than once. If this

element is specified, it is of type [Relationships](#). This element is ignored if the [DataSet](#) for this **TablixCell** is the same as the **DataSet** for each containing scope.

Following is the parent element of the **TablixCell.Relationships** element.

Parent elements
<a href="#">TablixCell</a>

The following is the XML Schema definition of the **TablixCell.Relationships** element.

```
<xsd:element name="Relationships" type="RelationshipsType" minOccurs="0" />
```

## 2.31 CellContents

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CellContents** element defines the contents of cells of a [Tablix](#). The **CellContents** element can contain 0 or 1 report item elements. If the **Top**, **Left**, **Height**, and **Width** grandchild elements exist, they are ignored, and their values are interpreted as follows:

**Top:** 0

**Left:** 0

**Height:** The same as the height of the containing object.

**Width:** The same as the width of the containing object.

The following are the parent and child elements of the **CellContents** element.

Parent elements
<a href="#">TablixCell</a>
<a href="#">TablixCornerCell</a>
<a href="#">TablixHeader</a>

Child elements
<a href="#">CellContents.Chart</a>
<a href="#">CellContents.ColSpan</a>
<a href="#">CellContents.CustomReportItem</a>
<a href="#">CellContents.GaugePanel</a>
<a href="#">CellContents.Image</a>
<a href="#">CellContents.Line</a>
<a href="#">CellContents.Map</a>
<a href="#">CellContents.Rectangle</a>



Child elements
<a href="#">CellContents.RowSpan</a>
<a href="#">CellContents.Subreport</a>
<a href="#">CellContents.Tablix</a>
<a href="#">CellContents.Textbox</a>

The following is the XML Schema definition of the **CellContents** element in RDL 2008/01.

```
<xsd:complexType name="CellContentsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="ColSpan" type="xsd:unsignedInt" minOccurs="0" maxOccurs="1" />
  </xsd:choice>
  <xsd:element name="RowSpan" type="xsd:unsignedInt" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Line" type="LineType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Rectangle" type="RectangleType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Textbox" type="TextboxType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Image" type="ImageType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Subreport" type="SubreportType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Chart" type="ChartType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="GaugePanel" type="GaugePanelType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="CustomReportItem" type="CustomReportItemType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Tablix" type="TablixType" minOccurs="0" maxOccurs="1" />
  <xsd:any namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **CellContents** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="CellContentsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="ColSpan" type="xsd:unsignedInt" minOccurs="0" maxOccurs="1" />
  </xsd:choice>
  <xsd:element name="RowSpan" type="xsd:unsignedInt" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Line" type="LineType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Rectangle" type="RectangleType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Textbox" type="TextboxType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Image" type="ImageType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Subreport" type="SubreportType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Chart" type="ChartType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="GaugePanel" type="GaugePanelType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Map" type="MapType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="CustomReportItem" type="CustomReportItemType" minOccurs="0" maxOccurs="1" />
  <xsd:element name="Tablix" type="TablixType" minOccurs="0" maxOccurs="1" />
  <xsd:any namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.31.1 CellContents.Chart

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CellContents.Chart** element specifies a [Chart](#) in a [Tablix](#) cell. This element is optional and MUST NOT be specified more than once. The **Top**, **Left**, **Height**, and **Width** child element values are interpreted as follows:

**Top:** 0

**Left:** 0

**Height:** The same as the height of the containing object.

**Width:** The same as the width of the containing object.

If the **CellContents.Chart** element is specified, it is of type **Chart**.

Following is the parent element of the **CellContents.Chart** element.

Parent elements
<a href="#">CellContents</a>

The following is the XML Schema definition of the **CellContents.Chart** element.

```
<xsd:element name="Chart" type="ChartType" minOccurs="0"
maxOccurs="1" />
```

### 2.31.2 CellContents.ColSpan

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CellContents.ColSpan** element specifies the number of consecutive columns that the parent [CellContents](#) element uses. The **CellContents.ColSpan** element is optional and MUST NOT be specified more than once.

The **CellContents.ColSpan** element is ignored for **CellContents** elements that have a [TablixHeader](#) element ancestor.

If the **CellContents.ColSpan** element is specified, its value MUST be an **UnsignedInt**. If this element is not specified, its value is interpreted as 1.

The sum of the **CellContents.ColSpan** values that are descendants of each [TablixRow](#) element MUST equal the number of [TablixColumn](#) elements of the [Tablix](#).

If the **CellContents.ColSpan** element is a descendant of a [TablixCell](#) element, its value MUST be 1 unless one of the following conditions is true:

- All columns within the span are defined by **static members**.
- All columns within the span are siblings of the same **dynamic member**.

The sum of the value of all peer **CellContents.ColSpan** elements MUST equal the number of columns of their ancestor.

Following is the parent element of the **CellContents.ColSpan** element.

Parent elements
<a href="#">CellContents</a>

The following is the XML Schema definition of the **CellContents.ColSpan** element.

```
<xsd:element name="ColSpan" type="xsd:unsignedInt" minOccurs="0" maxOccurs="1">
</xsd:element>
```

### 2.31.3 CellContents.CustomReportItem

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CellContents.CustomReportItem** element specifies a **custom report item** in a [Tablix](#) cell. This element is optional and MUST NOT be specified more than once. The **Top**, **Left**, **Height**, and **Width** child element values are interpreted as follows:

**Top:** 0

**Left:** 0

**Height:** The same as the height of the containing object.

**Width:** The same as the width of the containing object.

If this element is specified, it is of type [CustomReportItem](#).

Following is the parent element of the **CellContents.CustomReportItem** element.

Parent elements
<a href="#">CellContents</a>

The following is the XML Schema definition of the **CellContents.CustomReportItem** element.

```
<xsd:element name="CustomReportItem" type="CustomReportItemType" minOccurs="0"
maxOccurs="1" />
```

### 2.31.4 CellContents.GaugePanel

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CellContents.GaugePanel** element specifies a **gauge panel** in a [Tablix](#) cell. This element is optional and MUST NOT be specified more than once. The **Top**, **Left**, **Height**, and **Width** child element values are interpreted as follows:

**Top:** 0

**Left:** 0

**Height:** The same as the height of the containing object.

**Width:** The same as the width of the containing object.

If the **CellContents.GaugePanel** element is specified, it is of type [GaugePanel](#).

Following is the parent element of the **CellContents.GaugePanel** element.

Parent elements
<a href="#">CellContents</a>

The following is the XML Schema definition of the **CellContents.GaugePanel** element.

```
<xsd:element name="GaugePanel" type="GaugePanelType" minOccurs="0" maxOccurs="1" />
```

### 2.31.5 CellContents.Image

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CellContents.Image** element specifies an [Image](#) in a [Tablix](#) cell. This element is optional and MUST NOT be specified more than once. The **Top**, **Left**, **Height**, and **Width** child element values are interpreted as follows:

**Top:** 0

**Left:** 0

**Height:** The same as the height of the containing object.

**Width:** The same as the width of the containing object.

If the **CellContents.Image** element is specified, it is of type **Image**.

Following is the parent element of the **CellContents.Image** element.

Parent elements
<a href="#">CellContents</a>

The following is the XML Schema definition of the **CellContents.Image** element.

```
<xsd:element name="Image" type="ImageType" minOccurs="0" maxOccurs="1" />
```

### 2.31.6 CellContents.Line

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CellContents.Line** element specifies a graphical **line** in a [Tablix](#) cell. This element is optional and MUST NOT be specified more than once. The **Top**, **Left**, **Height**, and **Width** child element values are interpreted as follows:

**Top:** 0

**Left:** 0

**Height:** The same as the height of the containing object.

**Width:** The same as the width of the containing object.

If the **CellContents.Line** element is specified, it is of type [Line](#).

Following is the parent element of the **CellContents.Line** element.

Parent elements
<a href="#">CellContents</a>

The following is the XML Schema definition of the **CellContents.Line** element.

```
<xsd:element name="Line" type="LineType" minOccurs="0" maxOccurs="1" />
```

### 2.31.7 CellContents.Map

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **CellContents.Map** element specifies a **map** in a [Tablix](#) cell. This element is optional and MUST NOT be specified more than once. The **Top**, **Left**, **Height**, and **Width** child element values is interpreted as follows:

**Top:** 0

**Left:** 0

**Height:** The same as the height of the containing object.

**Width:** The same as the width of the containing object.

The **CellContents.Map** element is of type [Map](#).

Following is the parent element of the **CellContents.Map** element.

Parent elements
<a href="#">CellContents</a>

The following is the XML Schema definition of the **CellContents.Map** element.

```
<xsd:element name="Map" type="MapType" minOccurs="0" maxOccurs="1" />
```

### 2.31.8 CellContents.Rectangle

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CellContents.Rectangle** element specifies a **rectangle** in a [Tablix](#) cell. This element is optional and MUST NOT be specified more than once. The **Top**, **Left**, **Height**, and **Width** child element values are interpreted as follows:

**Top:** 0

**Left:** 0

**Height:** The same as the height of the containing object.

**Width:** The same as the width of the containing object.

If this element is specified, it is of type [Rectangle](#).

Following is the parent element of the **CellContents.Rectangle** element.

Parent elements
<a href="#">CellContents</a>

The following is the XML Schema definition of the **CellContents.Rectangle** element.

```
<xsd:element name="Rectangle" type="RectangleType" minOccurs="0" maxOccurs="1" />
```

### 2.31.9 CellContents.RowSpan

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CellContents.RowSpan** element specifies the number of consecutive rows that are used by the parent [CellContents](#) element. This element is optional and MUST NOT be specified more than once.

The **CellContents.RowSpan** element is ignored for **CellContents** elements that have a [TablixHeader](#) element ancestor.

If the **CellContents.RowSpan** element is specified, its value MUST be an **UnsignedInt**. If this element is not specified, its value is interpreted as 1. If this element has a [TablixCell](#) ancestor, the value of the **CellContents.RowSpan** element MUST be 1. The sum of the value of all peer **CellContents.RowSpan** elements MUST equal the number of rows of their ancestor.

Following is the parent element of the **CellContents.RowSpan** element.

Parent elements
<a href="#">CellContents</a>

The following is the XML Schema definition of the **CellContents.RowSpan** element.

```
<xsd:element name="RowSpan" type="xsd:unsignedInt" minOccurs="0" maxOccurs="1">  
</xsd:element>
```

### 2.31.10 CellContents.Subreport

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CellContents.Subreport** element specifies a subreport in a [Tablix](#) cell. This element is optional and MUST NOT be specified more than once. The **Top**, **Left**, **Height**, and **Width** child element values are interpreted as follows:

**Top:** 0

**Left:** 0

**Height:** The same as the height of the containing object.

**Width:** The same as the width of the containing object.

If the **CellContents.Subreport** element is specified, it is of type [Subreport](#).

Following is the parent element of the **CellContents.Subreport** element.

Parent elements
<a href="#">CellContents</a>

The following is the XML Schema definition of the **CellContents.Subreport** element.

```
<xsd:element name="Subreport" type="SubreportType" minOccurs="0" maxOccurs="1" />
```

### 2.31.11 CellContents.Tablix

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CellContents.Tablix** element specifies a [Tablix](#) within another tablix. This element is optional and MUST NOT be specified more than once. The **Top**, **Left**, **Height**, and **Width** child element values are interpreted as follows:

**Top:** 0

**Left:** 0

**Height:** The same as the height of the containing object.

**Width:** The same as the width of the containing object.

If the **CellContents.Tablix** element is specified, it is of type **Tablix**.

Following is the parent element of the **CellContents.Tablix** element.

Parent elements
<a href="#">CellContents</a>

The following is the XML Schema definition of the **CellContents.Tablix** element.

```
<xsd:element name="Tablix" type="TablixType" minOccurs="0" maxOccurs="1" />
```

### 2.31.12 CellContents.Textbox

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CellContents.Textbox** element specifies a [Textbox](#) in a [Tablix](#) cell. This element is optional and MUST NOT be specified more than once. The **Top**, **Left**, **Height**, and **Width** child element values are interpreted as follows:

**Top:** 0

**Left:** 0

**Height:** The same as the height of the containing object.

**Width:** The same as the width of the containing object.

If the **CellContents.Textbox** element is specified, it is of type **Textbox**.

Following is the parent element of the **CellContents.Textbox** element.

Parent elements
<a href="#">CellContents</a>

The following is the XML Schema definition of the **CellContents.Textbox** element.

```
<xsd:element name="Textbox" type="TextboxType" minOccurs="0" maxOccurs="1" />
```

## 2.32 TablixColumnHierarchy

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixColumnHierarchy** parent element defines the hierarchy of [TablixMember](#) elements for columns in a [Tablix](#). The cumulative size of [TablixHeader](#) leaf **members** (excluding leaves that have a [CellContents](#) element ancestor) MUST be equal.

"Cumulative size" for a **TablixHeader** element is defined as the sum of all [RdlSize](#) element values of the **TablixHeader** element and all of its ancestor **TablixHeader** elements.

The following are the parent and child elements of the **TablixColumnHierarchy** element.

Parent elements
<a href="#">Tablix</a>

Child elements
<a href="#">TablixColumnHierarchy.TablixMembers</a>

Applies to [RDL 2012/01](#)

The following is an additional child element of the **TablixColumnHierarchy** element.

Child elements
<a href="#">TablixColumnHierarchy.EnableDrilldown</a>

The following is the XML Schema definition of the **TablixColumnHierarchy** element in RDL 2008/01.

```
<xsd:complexType name="TablixHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TablixColumnHierarchy** element in RDL 2010/01 and RDL 2016/01.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections [5.5](#), [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:complexType name="TablixHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```



### 2.32.1 TablixColumnHierarchy.TablixMembers

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixColumnHierarchy.TablixMembers** element specifies **members** of a [TablixColumnHierarchy](#) element. The **TablixColumnHierarchy.TablixMembers** element MUST be specified exactly once and is of type [TablixMembers](#).

Following is the parent element of the **TablixColumnHierarchy.TablixMembers** element.

Parent elements
<a href="#">TablixColumnHierarchy</a>

The following is the XML Schema definition of the **TablixColumnHierarchy.TablixMembers** element.

```
<xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1" maxOccurs="1" />
```

### 2.32.2 TablixColumnHierarchy.EnableDrilldown

Applies to [RDL 2012/01](#)

The **TablixColumnHierarchy.EnableDrilldown** element specifies whether drilldown is enabled for the **member** hierarchy. If **TablixColumnHierarchy.EnableDrilldown** is specified, its value MUST be [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). A value of true specifies that the drilldown interactivity is enabled and that a portion of the hierarchy appears. The drilldown **filter** state determines which portion of the hierarchy appears. If the **TablixColumnHierarchy.EnableDrilldown** element is not specified, its value is interpreted as false.

Following is the parent element of the **TablixColumnHierarchy.EnableDrilldown** element.

Parent elements
<a href="#">TablixColumnHierarchy</a>

The following is the XML Schema definition of the **TablixColumnHierarchy.EnableDrilldown** element.

```
<xsd:element name="EnableDrilldown" type="xsd:boolean" />
```

## 2.33 TablixMembers

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixMembers** element defines a collection of [TablixMember](#) elements for a hierarchy.

The following are the parent and child elements of the **TablixMembers** element.

Parent elements
<a href="#">TablixColumnHierarchy</a>
TablixMember
<a href="#">TablixRowHierarchy</a>

<b>Child elements</b>
-----------------------

<a href="#">TablixMembers.TablixMember</a>
--

The following is the XML Schema definition of the **TablixMembers** element in RDL 2008/01.

```
<xsd:complexType name="TablixMembersType">
  <xsd:sequence minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixMember" type="TablixMemberType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TablixMembers** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="TablixMembersType">
  <xsd:sequence minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixMember" type="TablixMemberType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.33.1 TablixMembers.TablixMember

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixMembers.TablixMember** element specifies a [TablixMember](#) of a [TablixMembers](#) element. The **TablixMembers.TablixMember** element MUST be specified at least once. This element is of type **TablixMember**.

Following is the parent element of the **TablixMembers.TablixMember** element.

<b>Parent elements</b>
------------------------

<a href="#">TablixMembers</a>
-------------------------------

The following is the XML Schema definition of the **TablixMembers.TablixMember** element.

```
<xsd:element name="TablixMember" type="TablixMemberType" minOccurs="1"
  maxOccurs="unbounded" />
```

### 2.34 TablixMember

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixMember** parent element defines the structure of a [Tablix](#) hierarchy.

The following are the parent and child elements of the **TablixMember** element.

<b>Parent elements</b>
<a href="#">TablixMembers</a>

<b>Child elements</b>
<a href="#">TablixMember.CustomProperties</a>
<a href="#">TablixMember.DataElementName</a>
<a href="#">TablixMember.DataElementOutput</a>
<a href="#">TablixMember.FixedData</a>
<a href="#">TablixMember.Group</a>
<a href="#">TablixMember.HideIfNoRows</a>
<a href="#">TablixMember.KeepTogether</a>
<a href="#">TablixMember.KeepWithGroup</a>
<a href="#">TablixMember.RepeatOnNewPage</a>
<a href="#">TablixMember.SortExpressions</a>
<a href="#">TablixMember.TablixHeader</a>
<a href="#">TablixMember.TablixMembers</a>
<a href="#">TablixMember.Visibility</a>

The following is the XML Schema definition of the **TablixMember** element in RDL 2008/01.

```

<xsd:complexType name="TablixMemberType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="TablixHeader" type="TablixHeaderType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="FixedData" type="xsd:boolean" minOccurs="0" maxOccurs="1" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="HideIfNoRows" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="RepeatOnNewPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="KeepWithGroup" minOccurs="0" maxOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Before" />
          <xsd:enumeration value="After" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>

```

```

        <xsd:enumeration value="Auto" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **TablixMember** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="TablixMemberType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="TablixHeader" type="TablixHeaderType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="FixedData" type="xsd:boolean" minOccurs="0" maxOccurs="1" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="HideIfNoRows" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="RepeatOnNewPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="KeepWithGroup" minOccurs="0" maxOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Before" />
          <xsd:enumeration value="After" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.34.1 TablixMember.CustomProperties

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixMember.CustomProperties** element specifies custom information that is associated with a [TablixMember](#) element. The **TablixMember.CustomProperties** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [CustomProperties](#).

If the [Tablix.BandLayoutOptions](#) element is specified, the **TablixMember.CustomProperties** element is ignored.

Following is the parent element of the **TablixMember.CustomProperties** element.

Parent elements
TablixMember

The following is the XML Schema definition of the **TablixMember.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0"
maxOccurs="1" />
```

### 2.34.2 TablixMember.DataElementName

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixMember.DataElementName** element specifies the name to use for a **data element** of the [TablixMember](#) in a **data rendering**. The **TablixMember.DataElementName** element is optional and MUST NOT be specified more than once.

If the **TablixMember.DataElementName** element is specified, its value MUST be a [String](#) ([[XMLSCHEMA2/2](#)] section 3.2.1) that is a **CLS-compliant identifier** [[UTR15](#)] that is unique within the parent data element.

If the parent element is a **dynamic member** and if the **TablixMember.DataElementName** element is not specified, its value is interpreted as the value of the [Group.Name](#) attribute of the [Group](#) element sibling concatenated with "\_Collection".

If the parent element is a **static member** and if the **TablixMember.DataElementName** element is not specified, its value is interpreted as the value of the first descendant **TablixMember.DataElementName** element of the parent element's [TablixHeader](#) element.

If the value of the **TablixMember.DataElementName** element is not specified and cannot have a viable default, the value of this element is interpreted as "null".

Following is the parent element of the **TablixMember.DataElementName** element.

Parent elements
TablixMember

The following is the XML Schema definition of the **TablixMember.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.34.3 TablixMember.DataElementOutput

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixMember.DataElementOutput** element specifies the inclusion of a [TablixMember](#) in a **data rendering**. This element is optional and MUST NOT be specified more than once.

If this element is specified, its value MUST be one of the following:

**Auto** (default): If the parent element is a **dynamic member**, this value is interpreted as "Output".

**Output**: The **TablixMember** appears in the data rendering output.

**NoOutput:** The **TablixMember** does not appear in the data rendering output.

If the parent element is a **static member** and if the parent element has a [TablixHeader](#) element, this value is interpreted as "Output". Otherwise, this value is interpreted as "NoOutput".

If the **TablixMember.DataElementOutput** element is not specified, its value is interpreted as "Auto".

Following is the parent element of the **TablixMember.DataElementOutput** element.

Parent elements
TablixMember

The following is the XML Schema definition of the **TablixMember.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

#### 2.34.4 TablixMember.FixedData

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixMember.FixedData** element specifies whether a [TablixMember](#) element and its associated [TablixBody](#) element cells remain visible when the [Tablix](#) is partially scrolled off a page. The **TablixMember.FixedData** element is optional and MUST NOT be specified more than once.

If the **TablixMember.FixedData** element is specified, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). A value of true specifies that the **TablixMember** element and its associated **TablixBody** element cells remain visible. The value of the **TablixMember.FixedData** element is ignored if the parent **TablixMember** element has a **TablixMember** element ancestor in the tablix.

All **TablixMember** elements that have a **TablixMember.FixedData** element that has a value of true MUST be contiguous with all sibling **TablixMember** elements that have a **FixedData** element that has a value of true.

If any **TablixMember** element that has a [TablixRowHierarchy](#) element ancestor in the tablix has a **FixedData** element that has a value of true, the first **TablixMember** element that has a **TablixRowHierarchy** element ancestor in the tablix MUST have a **FixedData** element that has a value of true.

All **TablixMember** elements that have a child **FixedData** element value that resolves to true MUST have a common [TablixHierarchy](#) element ancestor in the tablix. This ancestor MUST be either a [TablixColumnHierarchy](#) element or a **TablixRowHierarchy** element.

If the tablix element ancestor has a child [Tablix.GroupsBeforeRowHeaders](#) element value that is greater than 0, and if the parent of this element is the first tablix column member, then the value of the **TablixMember.FixedData** element MUST be false.

If the **TablixMember.FixedData** element is not specified, its value is interpreted as false.

If the value of the **TablixMember.FixedData** element resolves to true and if the **TablixMember** element parent is a tablix column member that has no ancestor **TablixMember** element parent in the tablix, the sum of values of all [CellContents.ColSpan](#) element descendants for all **TablixMember** element peers MUST be equal to the quantity of all **TablixMember** element peers.

If the [Tablix.BandLayoutOptions](#) element is specified, the **TablixMember.FixedData** element is ignored.

Following is the parent element of the **TablixMember.FixedData** element.

Parent elements
TablixMember

The following is the XML Schema definition of the **TablixMember.FixedData** element.

```
<xsd:element name="FixedData" type="xsd:boolean" minOccurs="0" maxOccurs="1" />
```

### 2.34.5 TablixMember.Group

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixMember.Group** element specifies the group for a [TablixMember](#) element. The **TablixMember.Group** element is optional and MUST NOT be specified more than once.

If the **TablixMember.Group** element is specified, it is of type [Group](#). If a **TablixMember** element has a **detail member** ancestor, this element MUST NOT be specified.

Following is the parent element of the **TablixMember.Group** element.

Parent elements
TablixMember

The following is the XML Schema definition of the **TablixMember.Group** element.

```
<xsd:element name="Group" type="GroupType" minOccurs="0" maxOccurs="1" />
```

### 2.34.6 TablixMember.HideIfNoRows

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixMember.HideIfNoRows** element specifies whether a row or column that is associated with a static member is to be hidden if the tablix contains no rows of data. This element is optional and MUST NOT be specified more than once.

The **TablixMember.HideIfNoRows** element is ignored for nonstatic members. If this element is specified, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2).

If the tablix contains no rows of data and if the value of the **TablixMember.HideIfNoRows** element is true, the following conditions apply:

- Rows that are associated with the parent [TablixMember](#) are to be hidden.
- Sibling [Visibility](#) elements are ignored.

If the **TablixMember.HideIfNoRows** element is not specified, its value is interpreted as false.

If the [Tablix.BandLayoutOptions](#) element is specified, the **TablixMember.HideIfNoRows** element is ignored.

Following is the parent element of the **TablixMember.HideIfNoRows** element.

Parent elements
TablixMember

The following is the XML Schema definition of the **TablixMember.HideIfNoRows** element.

```
<xsd:element name="HideIfNoRows" type="xsd:boolean" minOccurs="0" />
```

### 2.34.7 TablixMember.KeepTogether

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixMember.KeepTogether** element specifies whether an attempt is made to render instances of a [TablixMember](#) on a single page. This element is optional and MUST NOT be specified more than once.

If the **TablixMember.KeepTogether** element is specified, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If the value of this element is true, instances of the **TablixMember** SHOULD<12> be rendered on a single page. If the **TablixMember.KeepTogether** element is not specified, its value is interpreted as false.

Following is the parent element of the **TablixMember.KeepTogether** element.

Parent elements
TablixMember

The following is the XML Schema definition of the **TablixMember.KeepTogether** element.

```
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
```

### 2.34.8 TablixMember.KeepWithGroup

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixMember.KeepWithGroup** element specifies whether an attempt is made to keep each rendered instance of a parent **static member** with the closest non-hidden instance of a preceding or subsequent sibling **dynamic member**. This element is optional and MUST NOT be specified more than once.

If the **TablixMember.KeepWithGroup** element is specified, its value MUST be one of the following:

**None** (default): The **TablixMember.KeepWithGroup** element is ignored.

**Before**: Rendered instances of the parent static member are kept with the closest preceding non-hidden sibling dynamic member.

**After**: Rendered instances of the parent static member are kept with the closest subsequent non-hidden sibling dynamic member.



The value of the **TablixMember.KeepWithGroup** element MUST be "None" if the parent element is a dynamic member or has a dynamic member descendant. The value of this element MUST be "None" if the parent element is a tablix column member.

If the value of the **TablixMember.KeepWithGroup** element is "Before", the value of this element for all sibling [TablixMember](#) elements between the parent of the **TablixMember.KeepWithGroup** element and the closest preceding non-hidden sibling dynamic member MUST be "Before".

If the value of the **TablixMember.KeepWithGroup** element is "After", the value of this element for all sibling **TablixMember** elements between the parent of the **TablixMember.KeepWithGroup** element and the closest subsequent non-hidden sibling dynamic member MUST be "After".

If the **TablixMember.KeepWithGroup** element is not specified, its value is interpreted as "None".

If the [Tablix.BandLayoutOptions](#) element is specified, the **TablixMember.KeepWithGroup** element is ignored.

Following is the parent element of the **TablixMember.KeepWithGroup** element.

Parent elements
TablixMember

The following is the XML Schema definition of the **TablixMember.KeepWithGroup** element.

```
<xsd:element name="KeepWithGroup" minOccurs="0" maxOccurs="1">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None" />
      <xsd:enumeration value="Before" />
      <xsd:enumeration value="After" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.34.9 TablixMember.RepeatOnNewPage

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TablixMember.RepeatOnNewPage** element specifies whether instances of rendered **static members** are repeated on every [Page](#) on which at least one complete instance of the associated **dynamic member** is rendered. An associated dynamic member is a dynamic member that is referenced by one of the following items:

- The [TablixMember.KeepWithGroup](#) element.
- A descendant of a referenced **TablixMember.KeepWithGroup** element whose child **TablixMember.KeepWithGroup** element has a value of "None".

The **TablixMember.RepeatOnNewPage** element is optional and MUST NOT be specified more than once. This element is ignored if the value of the sibling **TablixMember.KeepWithGroup** element is "None". If the **TablixMember.RepeatOnNewPage** element is specified, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2).

If the value of the **TablixMember.RepeatOnNewPage** element is true, instances of rendered static members are repeated on every page on which at least one complete instance of the associated dynamic member is rendered. If the parent element is a tablix column member, the value of this element MUST be false.

The value of the **TablixMember.RepeatOnNewPage** element for all sibling [TablixMember](#) elements between the parent of the **TablixMember.RepeatOnNewPage** element and the associated dynamic member MUST be the same.

If the **TablixMember.RepeatOnNewPage** element is not specified, its value is interpreted as false.

If the [Tablix.BandLayoutOptions](#) element is specified, the **TablixMember.RepeatOnNewPage** element is ignored.

Following is the parent element of the **TablixMember.RepeatOnNewPage** element.

Parent elements
TablixMember

The following is the XML Schema definition of the **TablixMember.RepeatOnNewPage** element.

```
<xsd:element name="RepeatOnNewPage" type="xsd:boolean" minOccurs="0" />
```

### 2.34.10 **TablixMember.SortExpressions**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixMember.SortExpressions** element specifies a collection of expressions that determine the order of the [TablixMember](#) data. The **TablixMember.SortExpressions** element is optional and MUST NOT be specified more than once.

The **TablixMember.SortExpressions** element MUST NOT be specified for static members. If this element is specified, it is of type [SortExpressions](#).

Following is the parent element of the **TablixMember.SortExpressions** element.

Parent elements
TablixMember

The following is the XML Schema definition of the **TablixMember.SortExpressions** element.

```
<xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" maxOccurs="1" />
```

### 2.34.11 **TablixMember.TablixHeader**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixMember.TablixHeader** element specifies a cell of a row header or a **column header**. This element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [TablixHeader](#).

If the [Tablix](#) has both a **TablixHeader** element with a [TablixRowHierarchy](#) element ancestor and a **TablixHeader** element with a [TablixColumnHierarchy](#) element ancestor, the tablix MUST also have a [TablixCorner](#) element descendant.

Following is the parent element of the **TablixMember.TablixHeader** element.

<b>Parent elements</b>
<a href="#">TablixMember</a>

The following is the XML Schema definition of the **TablixMember.TablixHeader** element.

```
<xsd:element name="TablixHeader" type="TablixHeaderType" minOccurs="0" maxOccurs="1" />
```

### 2.34.12 TablixMember.TablixMembers

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TablixMember.TablixMembers** element specifies a collection of [TablixMember](#) elements. The **TablixMember.TablixMembers** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [TablixMembers](#).

Following is the parent element of the **TablixMember.TablixMembers** element.

<b>Parent elements</b>
<a href="#">TablixMember</a>

The following is the XML Schema definition of the **TablixMember.TablixMembers** element.

```
<xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="0" maxOccurs="1" />
```

### 2.34.13 TablixMember.Visibility

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TablixMember.Visibility** element specifies the presentational presence of a [TablixMember](#) element. The **TablixMember.Visibility** element is optional and MUST NOT be specified more than once.

The **TablixMember.Visibility** element is ignored if a [Tablix](#) contains no rows of data and if the value of the [TablixMember.HideIfNoRows](#) element siblings is true. If the **TablixMember.Visibility** element is specified, it is of type [Visibility](#).

Following is the parent element of the **TablixMember.Visibility** element.

<b>Parent elements</b>
<a href="#">TablixMember</a>

The following is the XML Schema definition of the **TablixMember.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" maxOccurs="1" />
```

## 2.35 TablixHeader

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TablixHeader** parent element defines contents in both a **tablix** column group area and a tablix row group area.

The following are the parent and child elements of the **TablixHeader** element.

Parent elements
<a href="#">TablixMember</a>

Child elements
<a href="#">TablixHeader.CellContents</a>
<a href="#">TablixHeader.Size</a>

The following is the XML Schema definition of the **TablixHeader** element in RDL 2008/01.

```
<xsd:complexType name="TablixHeaderType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Size" type="SizeType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="CellContents" type="CellContentsType" minOccurs="1" maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TablixHeader** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="TablixHeaderType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Size" type="SizeType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="CellContents" type="CellContentsType" minOccurs="1" maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.35.1 TablixHeader.CellContents

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixHeader.CellContents** element specifies the contents of cells in a [TablixHeader](#). This element MUST be specified exactly once and is of type [CellContents](#). The [CellContents.ColSpan](#) and [CellContents.RowSpan](#) elements that are children of the **TablixHeader.CellContents** element is ignored.

Following is the parent element of the **TablixHeader.CellContents** element.

Parent elements
TablixHeader

The following is the XML Schema definition of the **TablixHeader.CellContents** element.

```
<xsd:element name="CellContents" type="CellContentsType" minOccurs="1" maxOccurs="1" />
```

## 2.35.2 TablixHeader.Size

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixHeader.Size** element specifies either the height or the width of a [TablixHeader](#) element. The **TablixHeader.Size** element MUST be specified exactly once and is of type [RdlSize](#). If the parent element is a tablix **column header**, the value of the [TablixRowHierarchy.TablixMembers](#) element specifies the width. If the parent element is a tablix row header, the value of the **RdlSize** specifies the height.

Following is the parent element of the **TablixHeader.Size** element.

Parent elements
<a href="#">TablixHeader</a>

The following is the XML Schema definition of the **TablixHeader.Size** element.

```
<xsd:element name="Size" type="SizeType" minOccurs="1" maxOccurs="1" />
```

## 2.36 TablixCorner

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixCorner** parent element defines the contents of the intersection of column group headers and row group headers.

The value of a [CellContents.ColSpan](#) element descendant in a **tablix**, decremented by 1, specifies the quantity of sequential sibling [TablixCornerCell](#) element descendants that MUST NOT have a child [TablixCornerCell.CellContents](#) element specified.

The value of a [CellContents.RowSpan](#) element descendant in the tablix, decremented by 1, specifies the quantity of sequential [TablixCornerRow](#) element siblings of the **TablixCornerRow** element ancestor of the RowSpan that MUST NOT have a child **TablixCornerCell.CellContents** element specified in the same positional order.

The following are the parent and child elements of the **TablixCorner** element.

Parent elements
<a href="#">Tablix</a>

Child elements
<a href="#">TablixCorner.TablixCornerRows</a>

The following is the XML Schema definition of the **TablixCorner** element in RDL 2008/01.

```
<xsd:complexType name="TablixCornerType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixCornerRows" type="TablixCornerRowsType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

```
</xsd:complexType>
```

The following is the XML Schema definition of the **TablixCorner** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="TablixCornerType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixCornerRows" type="TablixCornerRowsType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.36.1 TablixCorner.TablixCornerRows

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixCorner.TablixCornerRows** element specifies a collection of [TablixCornerCell](#) elements. This element MUST be specified exactly once. This element is of type [TablixCornerRows](#).

Following is the parent element of the **TablixCorner.TablixCornerRows** element.

Parent elements
<a href="#">TablixCorner</a>

The following is the XML Schema definition of the **TablixCorner.TablixCornerRows** element.

```
<xsd:element name="TablixCornerRows" type="TablixCornerRowsType" minOccurs="1"
  maxOccurs="1" />
```

### 2.37 TablixCornerRows

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixCornerRows** parent element defines the collection of rows in a [TablixCorner](#).

The following are the parent and child elements of the **TablixCornerRows** element.

Parent elements
<a href="#">TablixCorner</a>

Child elements
<a href="#">TablixCornerRows.TablixCornerRow</a>

The following is the XML Schema definition of the **TablixCornerRows** element in RDL 2008/01.

```
<xsd:complexType name="TablixCornerRowsType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
```

```

    <xsd:element name="TablixCornerRow" type="TablixCornerRowType" minOccurs="1"
      maxOccurs="unbounded">
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **TablixCornerRows** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="TablixCornerRowsType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixCornerRow" type="TablixCornerRowType" minOccurs="1"
      maxOccurs="unbounded">
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.37.1 TablixCornerRows.TablixCornerRow

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixCornerRows.TablixCornerRow** element specifies a row in a [TablixCorner](#). The **TablixCornerRows.TablixCornerRow** element is of type [TablixCornerRow](#).

If a **TablixCorner** element is specified, the quantity of descendant **TablixCornerRow** elements in the [Tablix](#) MUST equal the quantity of unique cumulative heights for each tablix column member of the tablix. "Cumulative height" for a tablix column member is defined as the sum of the values of all [TablixHeader.Size](#) elements that are descendants of the tablix column member.

Following is the parent element of the **TablixCornerRows.TablixCornerRow** element.

<b>Parent elements</b>
<a href="#">TablixCornerRows</a>

The following is the XML Schema definition of the **TablixCornerRows.TablixCornerRow** element.

```

<xsd:element name="TablixCornerRow" type="TablixCornerRowType" minOccurs="1"
  maxOccurs="unbounded">
</xsd:element>

```

### 2.38 TablixCornerRow

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixCornerRow** element defines a row in a [TablixCorner](#).

The following are the parent and child elements of the **TablixCornerRow** element.

<b>Parent elements</b>
<a href="#">TablixCornerRows</a>

<b>Child elements</b>
-----------------------

<a href="#">TablixCornerRow.TablixCornerCell</a>
--

The following is the XML Schema definition of the **TablixCornerRow** element in RDL 2008/01.

```
<xsd:complexType name="TablixCornerRowType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="TablixCornerCell" type="TablixCornerCellType" minOccurs="0"
      maxOccurs="unbounded">
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TablixCornerRow** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="TablixCornerRowType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="TablixCornerCell" type="TablixCornerCellType" minOccurs="0"
      maxOccurs="unbounded">
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.38.1 TablixCornerRow.TablixCornerCell

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixCornerRow.TablixCornerCell** element specifies a cell in a row in a [TablixCorner](#). The **TablixCornerRow.TablixCornerCell** element is of type [TablixCornerCell](#).

If a [TablixCornerRow](#) element is specified, the quantity of descendant **TablixCornerCell** elements in the [Tablix](#) MUST equal the quantity of unique cumulative widths for each tablix **row member** of the tablix. "Cumulative width" for a tablix row member is defined as the sum of the values of all [TablixHeader.Size](#) elements that are descendants of the tablix row member.

Following is the parent element of the **TablixCornerRow.TablixCornerCell** element.

<b>Parent elements</b>
------------------------

<a href="#">TablixCornerRow</a>
---------------------------------

The following is the XML Schema definition of the **TablixCornerRow.TablixCornerCell** element.

```
<xsd:element name="TablixCornerCell" type="TablixCornerCellType" minOccurs="0"
  maxOccurs="unbounded">
</xsd:element>
```



## 2.39 TablixCornerCell

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixCornerCell** element defines a cell of a [TablixCornerRow](#).

The following are the parent and child elements of the **TablixCornerCell** element.

Parent elements
<a href="#">TablixCornerRow</a>

Child elements
<a href="#">TablixCornerCell.CellContents</a>

The following is the XML Schema definition of the **TablixCornerCell** element in RDL 2008/01.

```
<xsd:complexType name="TablixCornerCellType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="CellContents" type="CellContentsType" minOccurs="0"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TablixCornerCell** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="TablixCornerCellType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="CellContents" type="CellContentsType" minOccurs="0"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.39.1 TablixCornerCell.CellContents

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixCornerCell.CellContents** element specifies the contents of cells in a [TablixCornerCell](#).

The **TablixCornerCell.CellContents** element MUST NOT be specified if the position in the corner is covered by a span from another cell. Otherwise, this element MUST be specified exactly once.

The following rules determine whether the position in the corner is covered by a span from another cell:

- The value of the [CellContents.ColSpan](#) child element, decremented by 1, specifies the quantity of sequential sibling **TablixCornerCell** elements that MUST NOT have a child **TablixCornerCell.CellContents** element specified.
- The value of the [CellContents.RowSpan](#) child element, decremented by 1, specifies the quantity of sequential [TablixCornerRow](#) element siblings to the ancestor **TablixCornerRow** element that

MUST NOT have a child **TablixCornerCell.CellContents** element specified in the same positional order.

If the **TablixCornerCell.CellContents** element is specified, it is of type [CellContents](#).

Following is the parent element of the **TablixCornerCell.CellContents** element.

Parent elements
TablixCornerCell

The following is the XML Schema definition of the **TablixCornerCell.CellContents** element.

```
<xsd:element name="CellContents" type="CellContentsType" minOccurs="0" maxOccurs="1" />
```

## 2.40 TablixHierarchy

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TablixHierarchy** parent element defines the hierarchy of [TablixMember](#) elements for a [Tablix](#). The cumulative size of [TablixHeader](#) leaf **members** (excluding leaves that have a [CellContents](#) element ancestor) MUST be equal.

"Cumulative size" for a **TablixHeader** element is defined as the sum of all [RdlSize](#) values of the **TablixHeader** element and all of its ancestor **TablixHeader** elements.

The following are the parent and child elements of the **TablixHierarchy** element.

Parent elements
Tablix

Child elements
<a href="#">TablixHierarchy.TablixMembers</a>

The following is the XML Schema definition of the **TablixHierarchy** element in RDL 2008/01.

```
<xsd:complexType name="TablixHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TablixHierarchy** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="TablixHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
</xsd:complexType>
```

```

</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.40.1 TablixHierarchy.TablixMembers

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TablixHierarchy.TablixMembers** element specifies members of a [TablixHierarchy](#) element. The **TablixHierarchy.TablixMembers** element MUST be specified exactly once and is of type [TablixMembers](#).

Following is the parent element of the **TablixHierarchy.TablixMembers** element.

Parent elements
<a href="#">TablixHierarchy</a>

The following is the XML Schema definition of the **TablixHierarchy.TablixMembers** element.

```

<xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1"
maxOccurs="1" />

```

### 2.41 TablixRowHierarchy

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TablixRowHierarchy** parent element defines the hierarchy of [TablixMember](#) elements for rows in a [Tablix](#). The cumulative size of [TablixHeader](#) leaf **members** (excluding leaves that have a [CellContents](#) element ancestor) MUST be equal.

"Cumulative size" for a **TablixHeader** element is defined as the sum of all [RdlSize](#) element values of the **TablixHeader** element and all of its ancestor **TablixHeader** elements.

The following are the parent and child elements of the **TablixRowHierarchy** element.

Parent elements
<a href="#">Tablix</a>

Child elements
<a href="#">TablixRowHierarchy.TablixMembers</a>

*Applies to [RDL 2012/01](#)*

The following is an additional child element of the **TablixRowHierarchy** element.

Child elements
<a href="#">TablixRowHierarchy.EnableDrilldown</a>

The following is the XML Schema definition of the **TablixRowHierarchy** element in RDL 2008/01.

```

<xsd:complexType name="TablixHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **TablixRowHierarchy** element in RDL 2010/01 and RDL 2016/01.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections [5.5](#), [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```

<xsd:complexType name="TablixHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.41.1 TablixRowHierarchy.TablixMembers

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TablixRowHierarchy.TablixMembers** element specifies members of a [TablixRowHierarchy](#) element. The **TablixRowHierarchy.TablixMembers** element MUST be specified exactly once and is of type [TablixMembers](#).

Following is the parent element of the **TablixRowHierarchy.TablixMembers** element.

<b>Parent elements</b>
<a href="#">TablixHierarchy</a>

The following is the XML Schema definition of the **TablixRowHierarchy.TablixMembers** element.

```

<xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1"
  maxOccurs="1" />

```

### 2.41.2 TablixRowHierarchy.EnableDrilldown

**Applies to** [RDL 2012/01](#)

The **TablixRowHierarchy.EnableDrilldown** element specifies whether drilldown is enabled for the member hierarchy. If **TablixRowHierarchy.EnableDrilldown** is specified, its value MUST be [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). A value of true specifies that the drilldown interactivity is enabled and that a portion of the hierarchy appears. The drilldown **filter** state determines which portion of the hierarchy appears. If the **TablixRowHierarchy.EnableDrilldown** element is not specified, its value is interpreted as false.

Following is the parent element of the **TablixRowHierarchy.EnableDrilldown** element.

Parent elements
<a href="#">TablixRowHierarchy</a>

The following is the XML Schema definition of the **TablixRowHierarchy.EnableDrilldown** element.

```
<xsd:element name ="EnableDrilldown" type="xsd:boolean" />
```

## 2.42 BandLayoutOptions

**Applies to** [RDL 2011/01](#)

The **BandLayoutOptions** element specifies layout options for a [Tablix](#) when it is displayed as a band.

The following are the parent and child elements of the **BandLayoutOptions** element.

Parent elements
<a href="#">Tablix</a>

Child elements
<a href="#">RowCount</a>
<a href="#">ColumnCount</a>
<a href="#">Coverflow</a>
<a href="#">PlayAxis</a>
<a href="#">Tabstrip</a>

The following is the XML Schema definition of the **BandLayoutOptions** element.

```
<xsd:element name ="BandLayoutOptions">
  <xsd:complexType>
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="RowCount" minOccurs="0" maxOccurs="1">
        <xsd:simpleType>
          <xsd:restriction base="xsd:unsignedInt">
            <xsd:minInclusive value="1"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="ColumnCount" minOccurs="0" maxOccurs="1">
        <xsd:simpleType>
          <xsd:restriction base="xsd:unsignedInt">
            <xsd:minInclusive value="1"/>
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="Coverflow" type="CoverflowType" minOccurs="0"/>
      <xsd:element name="PlayAxis" type="PlayAxisType" minOccurs="0"/>
      <xsd:element name="Tabstrip" type="TabstripType" minOccurs="0"/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

```

    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
</xsd:element>

```

### 2.42.1 BandLayoutOptions.RowCount

**Applies to** [RDL 2011/01](#)

The **BandLayoutOptions.RowCount** element specifies the number of [Tablix](#) rows to be displayed on a layout per **band sheet**. The **BandLayoutOptions.RowCount** element is optional.

If this element is specified, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) and MUST be greater than zero. If this element is not present, its value is interpreted as 1.

The following is the parent of the **BandLayoutOptions.RowCount** element.

<b>Parent elements</b>
<a href="#">BandLayoutOptions</a>

The following is the XML Schema definition of the **BandLayoutOptions.RowCount** element.

```

<xsd:element name="RowCount" minOccurs="0" maxOccurs="1">
  <xsd:simpleType>
    <xsd:restriction base="xsd:unsignedInt">
      <xsd:minInclusive value="1"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

```

### 2.42.2 BandLayoutOptions.ColumnCount

**Applies to** [RDL 2011/01](#)

The **BandLayoutOptions.ColumnCount** element specifies the number of [Tablix](#) columns to be displayed on a layout per **band sheet**. The **BandLayoutOptions.ColumnCount** element is optional.

If this element is specified, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) and MUST be greater than zero. If this element is not present, its value is interpreted as 1.

The following is the parent of the **BandLayoutOptions.ColumnCount** element.

<b>Parent elements</b>
<a href="#">BandLayoutOptions</a>

The following is the XML Schema definition of the **BandLayoutOptions.ColumnCount** element.

```

<xsd:element name="ColumnCount" minOccurs="0" maxOccurs="1">
  <xsd:simpleType>
    <xsd:restriction base="xsd:unsignedInt">
      <xsd:minInclusive value="1"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

```

### 2.42.3 BandLayoutOptions.Coverflow

*Applies to [RDL 2011/01](#)*

The **BandLayoutOptions.Coverflow** element specifies a navigation report item for a band with a **coverflow** layout. The **BandLayoutOptions.Coverflow** element is optional. This element is of type [Coverflow](#).

This element MUST NOT be specified if either [BandLayoutOptions.PlayAxis](#) or [BandLayoutOptions.Tabstrip](#) element is present.

The following is the parent element of the **BandLayoutOptions.Coverflow** element.

Parent elements
<a href="#">BandLayoutOptions</a>

The following is the XML Schema definition of the **BandLayoutOptions.Coverflow** element.

```
<xsd:element name="Coverflow" type="CoverflowType" minOccurs="0"/>
```

### 2.42.4 BandLayoutOptions.PlayAxis

*Applies to [RDL 2011/01](#)*

The **BandLayoutOptions.PlayAxis** element specifies a navigation report item for a band with a **play axis**. This element is optional. This element is of type [PlayAxis](#).

This element MUST NOT be specified if either [BandLayoutOptions.Coverflow](#) or [BandLayoutOptions.Tabstrip](#) element is present.

The following is the parent element of the **BandLayoutOptions.PlayAxis** element.

Parent elements
<a href="#">BandLayoutOptions</a>

The following is the XML Schema definition of the **BandLayoutOptions.PlayAxis** element.

```
<xsd:element name="PlayAxis" type="PlayAxisType" minOccurs="0"/>
```

### 2.42.5 BandLayoutOptions.Tabstrip

*Applies to [RDL 2011/01](#)*

The **BandLayoutOptions.Tabstrip** element specifies a navigation report item for a band with a **tabstrip** layout. This element is optional. This element is of type [Tabstrip](#).

This element MUST NOT be specified if either [BandLayoutOptions.Coverflow](#) or [BandLayoutOptions.PlayAxis](#) element is present.

The following is the parent element of the **BandLayoutOptions.Tabstrip** element.

<b>Parent elements</b>
------------------------

<a href="#">BandLayoutOptions</a>
-----------------------------------

The following is the XML Schema definition of the **BandLayoutOptions.Tabstrip** element.

```
<xsd:element name="Tabstrip" type="TabstripType" minOccurs="0"/>
```

## 2.43 Coverflow

**Applies to** [RDL 2011/01](#)

The **Coverflow** element specifies a report item or a reference to a report item to be used for navigation in a band with a **coverflow** layout.

The following are the parent and child elements of the **Coverflow** element.

<b>Parent elements</b>
------------------------

<a href="#">BandLayoutOptions</a>
-----------------------------------

<b>Child elements</b>
-----------------------

<a href="#">NavigationItem</a>
--------------------------------

<a href="#">Slider</a>
------------------------

The following is the XML Schema definition of the **Coverflow** element.

```
<xsd:complexType name="CoverflowType">  
  <xsd:choice minOccurs="1" maxOccurs="unbounded">  
    <xsd:element name="NavigationItem" type="NavigationItemType" minOccurs="0"  
      maxOccurs="1"/>  
    <xsd:element name="Slider" type="SliderType" minOccurs="0" maxOccurs="1"/>  
    <xsd:any namespace="##other" processContents="lax" />  
  </xsd:choice>  
  <xsd:anyAttribute namespace="##other" processContents="lax" />  
</xsd:complexType>
```

### 2.43.1 Coverflow.NavigationItem

**Applies to** [RDL 2011/01](#)

The **Coverflow.NavigationItem** element specifies a navigation report item or a reference to a navigation report item reference for a band. This element is optional. This element is of type [NavigationItem](#).

The following is the parent element of the **Coverflow.NavigationItem** element.



<b>Parent elements</b>
------------------------

<a href="#">Coverflow</a>
---------------------------

The following is the XML Schema definition of the Coverflow.NavigationItem element.

```
<xsd:element name="NavigationItem" type="NavigationItemType" minOccurs="0" maxOccurs="1"/>
```

## 2.43.2 Coverflow.Slider

**Applies to** [RDL 2011/01](#)

The **Coverflow.Slider** element specifies the **slider metadata** for band navigation. This element is optional. This element is of type [Slider](#).

<b>Parent elements</b>
------------------------

<a href="#">Coverflow</a>
---------------------------

The following is the XML Schema definition of the **Coverflow.Slider** element.

```
<xsd:element name="Slider" type="SliderType" minOccurs="0" maxOccurs="1"/>
```

## 2.44 Tabstrip

**Applies to** [RDL 2011/01](#)

The **Tabstrip** element specifies a report item or a reference to a report item to be used for navigation in a band with a **tabstrip** layout.

The following are the parent and child elements of the **Tabstrip** element.

<b>Parent elements</b>
------------------------

<a href="#">BandLayoutOptions</a>
-----------------------------------

<b>Child elements</b>
-----------------------

<a href="#">NavigationItem</a>
--------------------------------

<a href="#">Slider</a>
------------------------

The following is the XML Schema definition of the **Tabstrip** element.

```
<xsd:complexType name="TabstripType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="NavigationItem" type="NavigationItemType" minOccurs="0"
      maxOccurs="1"/>
    <xsd:element name="Slider" type="SliderType" minOccurs="0" maxOccurs="1"/>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.44.1 Tabstrip.NavigationItem

*Applies to [RDL 2011/01](#)*

The **Tabstrip.NavigationItem** element specifies a navigation report item or a reference to a navigation report item for a band. This element is optional. This element is of type [NavigationItem](#).

The following is the parent element of the **Tabstrip.NavigationItem** element.

Parent elements
<a href="#">Tabstrip</a>

The following is the XML Schema definition of the **Tabstrip.NavigationItem** element.

```
<xsd:element name="NavigationItem" type="NavigationItemType" minOccurs="0" maxOccurs="1"/>
```

## 2.44.2 Tabstrip.Slider

*Applies to [RDL 2011/01](#)*

The **Tabstrip.Slider** element specifies the **slider metadata** for band navigation. This element is optional. This element is of type [Slider](#).

The following is the parent element of the **Tabstrip.Slider** element.

Parent elements
<a href="#">Tabstrip</a>

The following is the XML Schema definition of the **Tabstrip.Slider** element.

```
<xsd:element name="Slider" type="SliderType" minOccurs="0" maxOccurs="1"/>
```

## 2.45 PlayAxis

*Applies to [RDL 2011/01](#)*

The **PlayAxis** element specifies a report item or a reference to a report item to be used for navigation in a band with a **play axis** layout.

The following are the parent and child elements of the **PlayAxis** element.

Parent elements
<a href="#">BandLayoutOptions</a>

Child elements
<a href="#">Slider</a>
<a href="#">DockingOption</a>

The following is the XML Schema definition of the **PlayAxis** element.

```

<xsd:complexType name="PlayAxisType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Slider" type="SliderType" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="DockingOption" minOccurs="0" maxOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Top"/>
          <xsd:enumeration value="Bottom"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.45.1 PlayAxis.Slider

**Applies to** [RDL 2011/01](#)

The **Tabstrip.Slider** element specifies the **slider metadata** for band navigation. This element is optional. This element is of type [Slider](#).

The following is the parent element of the **PlayAxis.Slider** element.

Parent elements
<a href="#">PlayAxis</a>

The following is the XML Schema definition of the **PlayAxis.Slider** element.

```

<xsd:element name="Slider" type="SliderType" minOccurs="0" maxOccurs="1"/>

```

### 2.45.2 PlayAxis.DockingOption

**Applies to** [RDL 2011/01](#)

The **PlayAxis.DockingOption** element specifies the docking option for the band **slider**. This element is optional. If this element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**Top:** Slider is at the top of the band.

**Bottom:** Slider is at the bottom of the band.

If the **PlayAxis.DockingOption** element is not specified, its value is interpreted as "Bottom".

The following is the parent element of the **PlayAxis.DockingOption** element.

Parent elements
<a href="#">PlayAxis</a>

The following is the XML Schema definition of the **PlayAxis.DockingOption** element.

```

<xsd:element name="DockingOption" minOccurs="0" maxOccurs="1">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">

```

```

        <xsd:enumeration value="Top"/>
        <xsd:enumeration value="Bottom"/>
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>

```

## 2.46 NavigationItem

**Applies to** [RDL 2011/01](#)

The **NavigationItem** element specifies a report item or a reference to a report item to be used for navigation in a band.

The following are the parent and child elements of the **NavigationItem** element.

Parent elements
<a href="#">Coverflow</a>
<a href="#">Tabstrip</a>

Child elements
<a href="#">ReportItemReference</a>
<a href="#">ReportItem</a>

The following is the XML Schema definition of the **NavigationItem** element.

```

<xsd:complexType name="NavigationItemType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ReportItemReference" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ReportItem" type="rdl2010:ReportItemsType" minOccurs="0"
      maxOccurs="1"/>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.46.1 NavigationItem.ReportItemReference

**Applies to** [RDL 2011/01](#)

The **NavigationItem.ReportItemReference** element specifies the name of the report item to be used for navigation in the band. This element is optional. If this element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is the name of a report item in the body of the band tablix.

The referred report item MUST NOT be inside a grouping of a nested tablix of a band tablix.

If a nested tablix has its [Tablix.Filters](#) element specified, the value of **NavigationItem.ReportItemReference** MUST NOT be any of the report item names inside that nested tablix.

If [NavigationItem.ReportItem](#) is specified, this element MUST NOT be specified.

The following is the parent element of the **NavigationItem.ReportItemReference** element.

Parent elements
<a href="#">NavigationItem</a>

The following is the XML Schema definition of the **NavigationItem.ReportItemReference** element.

```
<xsd:element name="ReportItemReference" type="xsd:string" minOccurs="0" maxOccurs="1"/>
```

## 2.46.2 NavigationItem.ReportItem

**Applies to** [RDL 2011/01](#)

The **NavigationItem.ReportItem** element specifies the **report item** to be used for navigation in the band. This element is optional. This element is on type **ReportItem**.

If [NavigationItem.ReportItemReference](#) is specified, this element MUST NOT be specified.

The following is the parent element of the **NavigationItem.ReportItem** element.

Parent elements
<a href="#">NavigationItem</a>

The following is the XML Schema definition of the **NavigationItem.ReportItem** element.

```
<xsd:element name="ReportItem" type="rdl2010:ReportItemsType" minOccurs="0" maxOccurs="1"/>
```

## 2.47 Slider

**Applies to** [RDL 2011/01](#)

The **Slider** element specifies the **slider metadata** for band navigation.

The following are the parent and child elements of the **Slider** element.

Parent elements
<a href="#">Coverflow</a>
<a href="#">Tabstrip</a>
<a href="#">PlayAxis</a>

Child elements
<a href="#">Hidden</a>
<a href="#">LabelData</a>

The following is the XML Schema definition of the **Slider** element.

```
<xsd:complexType name="SliderType">
```

```

<xsd:choice minOccurs="1" maxOccurs="unbounded">
  <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" maxOccurs="1"/>
  <xsd:element name="LabelData" type="LabelDataType" minOccurs="0" maxOccurs="1"/>
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.47.1 Slider.Hidden

**Applies to** [RDL 2011/01](#)

The **Slider.Hidden** element specifies whether the **slider** is hidden in the band.

This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false.

The following is the parent element of the **Slider.Hidden** element.

Parent elements
<a href="#">Slider</a>

The following is the XML Schema definition of the **Slider.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" maxOccurs="1"/>
```

### 2.47.2 Slider.LabelData

**Applies to** [RDL 2011/01](#)

The **Slider.LabelData** element specifies the **dataset** to populate the slider metadata for a band. This element is optional. This element is of type [LabelData](#).

The following is the parent element of the **Slider.LabelData** element.

Parent elements
<a href="#">Slider</a>

The following is the XML Schema definition of the **Slider.LabelData** element.

```
<xsd:element name="LabelData" type="LabelDataType" minOccurs="0" maxOccurs="1"/>
```

## 2.48 LabelData

**Applies to** [RDL 2011/01](#)

The **LabelData** element specifies the **dataset** to populate a **slider metadata** for a band.

The following are the parent and child elements of the **LabelData** element.

Parent elements
<a href="#">Slider</a>

Child elements
<a href="#">DataSetName</a>
<a href="#">Key</a>
<a href="#">Label</a>

**Applies to** [RDL 2013/01](#)

Child elements
<a href="#">KeyFields</a>

The following is the XML Schema definition of the **LabelData** element.

**Note** The following XSD represents the micro-versioned RDL 2011/01 schema only. The micro-versioned RDL 2013/01 schema is provided in section 5.7. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:complexType name="LabelDataType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="Key" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="Label" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.48.1 LabelData.DataSetName

**Applies to** [RDL 2011/01](#)

The **LabelData.DataSetName** element specifies the name of the [DataSet](#) to use for a slider metadata. The **LabelData.DataSetName** element is optional and MUST NOT be specified more than once.

If this element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the [Dataset.Name](#) attribute values of the report.

If there is more than one **DataSet** for the containing report, the value of the **LabelData.DataSetName** element MUST be specified.

If no value is specified for the **LabelData.DataSetName** element and if the containing report has only one **DataSet** specified, the value of **LabelData.DataSetName** is interpreted as being the value of that **DataSet.Name** attribute.

The following is the parent element of the **LabelData.DataSetName** element.

Parent elements
-----------------

<a href="#">LabelData</a>
---------------------------

The following is the XML Schema definition of the **LabelData.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" maxOccurs="1"/>
```

## 2.48.2 LabelData.Key

**Applies to** [RDL 2011/01](#)

The **LabelData.Key** element specifies the name of the [DataSet](#) Field to be matched with a band ([Tablix](#)) **group expression** value. The **LabelData.Key** element is optional. If this element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the [Field.Name](#) attribute values of the **DataSet** for slider metadata.

The following is the parent element of the **LabelData.Key** element.

Parent elements
-----------------

<a href="#">LabelData</a>
---------------------------

The following is the XML Schema definition of the **LabelData.Key** element.

```
<xsd:element name="Key" type="xsd:string" minOccurs="0" maxOccurs="1"/>
```

## 2.48.3 LabelData.KeyFields

**Applies to** [RDL 2013/01](#)

The **LabelData.KeyFields** element supersedes the [LabelData.Key](#) element of [RDL 2011/01](#). The **LabelData.KeyFields** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type **KeyFields**. The **LabelData.KeyFields** element is a collection of **Key** elements. The value of each **Key** MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the [Field.Name](#) attribute values of the **DataSet** for slider metadata.

Either the **LabelData.KeyFields** element or the **LabelData.Key** element of RDL 2011/01 MUST be specified. If both elements are specified, **LabelData.Key** is ignored.

The following is the parent element of the **LabelData.KeyFields** element.

Parent elements
-----------------

<a href="#">LabelData</a>
---------------------------

The following is the XML Schema definition of the **LabelData.KeyFields** element.

```
<xsd:element name="KeyFields">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Key" type="xsd:string" minOccurs="1" maxOccurs="unbounded" />
      <xsd:any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
</xsd:element>
```



```
</xsd:complexType>
</xsd:element>
```

## 2.48.4 LabelData.Label

**Applies to** [RDL 2011/01](#)

The **LabelData.Label** element specifies the name of the [DataSet](#) Field to be shown as navigation label for the band ([Tablix](#)). The **LabelData.Label** element is optional. If this element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the [Field.Name](#) attribute values of the **DataSet** for slider metadata.

The following is the parent element of the **LabelData.Label** element.

Parent elements
<a href="#">LabelData</a>

The following is the XML Schema definition of the **LabelData.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" maxOccurs="1"/>
```

## 2.49 List

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List** element specifies a region that repeats with each group in the **rowset** or each row in the rowset if no **group expression** is defined.

The following are the parent elements, attributes, and child elements of the **List** element.

Parent elements
<a href="#">ReportItems</a>
<a href="#">CustomReportItem.AltReportItem</a>

Attributes
<a href="#">List.Name</a>

Child elements
<a href="#">List.Style</a>
<a href="#">List.Action</a>
<a href="#">List.LinkToChild</a>
<a href="#">List.Bookmark</a>

Child elements
<a href="#">List.CustomProperties</a>
<a href="#">List.DataElementName</a>
<a href="#">List.DataElementOutput</a>
<a href="#">List.Label</a>
<a href="#">List.Height</a>
<a href="#">List.Left</a>
<a href="#">List.RepeatWith</a>
<a href="#">List.ToolTip</a>
<a href="#">List.Top</a>
<a href="#">List.Visibility</a>
<a href="#">List.Width</a>
<a href="#">List.ZIndex</a>
<a href="#">List.KeepTogether</a>
<a href="#">List.NoRows</a>
<a href="#">List.PageBreakAtEnd</a>
<a href="#">List.PageBreakAtStart</a>
<a href="#">List.DataSetName</a>
<a href="#">List.Filters</a>
<a href="#">List.DataInstanceElementOutput</a>
<a href="#">List.DataInstanceName</a>
<a href="#">List.FillPage</a>
<a href="#">List.Grouping</a>
<a href="#">List.ReportItems</a>
<a href="#">List.Sorting</a>

The following is the XML Schema definition of the **List** element.

```

<xsd:complexType name="ListType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>

```

```

<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
<xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:element name="Grouping" type="GroupingType" minOccurs="0" />
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
<xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
<xsd:element name="FillPage" type="xsd:boolean" minOccurs="0" />
<xsd:element name="DataInstanceName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataInstanceElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

## 2.49.1 List.Name

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.Name** attribute specifies a unique identifier for a [List](#). This attribute **MUST** be specified. The value of the **List.Name** attribute **MUST** be a [NormalizedString](#) that is a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#). This value **MUST** be unique among the set of all report item and **scope** names.

Following is the parent element of the **List.Name** attribute.

Parent elements
List

The following is the XML Schema definition of the **List.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

## 2.49.2 List.Style

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.Style** element specifies the style information for a [List](#). This element is optional and is of type [Style](#).

Following is the parent element of the **List.Style** element.

Parent elements
<a href="#">List</a>

The following is the XML Schema definition of the **List.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.49.3 List.Action

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.Action** element is ignored.

Following is the parent element of the **List.Action** element.

Parent elements
<a href="#">List</a>

The following is the XML Schema definition of the **List.Action** element.

```
<xsd:element name="Action" type="ActionType" minOccurs="0" />
```

## 2.49.4 List.LinkToChild

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.LinkToChild** element is ignored.

Following is the parent element of the **List.LinkToChild** element.

Parent elements
<a href="#">List</a>

The following is the XML Schema definition of the **List.LinkToChild** element.

```
<xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
```

## 2.49.5 List.Bookmark

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.Bookmark** element specifies a **bookmark** for a [List](#) that can be linked to via a bookmark action. This element is optional. If the **List.Bookmark** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **List.Bookmark** element.

Parent elements
List

The following is the XML Schema definition of the **List.Bookmark** element.

```
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
```

## 2.49.6 List.CustomProperties

*Applies to [RDL 2005/01](#)*

The **List.CustomProperties** element specifies extended information in a collection of name/value pairs. This element is optional and is of type [CustomProperties](#).

Following is the parent element of the **List.CustomProperties** element.

Parent elements
<a href="#">List</a>

The following is the XML Schema definition of the **List.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

In [RDL 2003/10](#), the equivalent element of **List.CustomProperties** is **List.Custom**, which is of type [Custom](#).

## 2.49.7 List.DataElementName

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **List.DataElementName** element specifies the name to use for the **data element** of a **list** in a **data rendering**. This element is optional. If this element is present, its value MUST be a case-sensitive **CLS-compliant identifier** [UTR15]. If this element is not present, its value is interpreted as the value of the **Name** attribute of the [List](#) element.

Following is the parent element of the **List.DataElementName** element.

Parent elements
List

The following is the XML Schema definition of the **List.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

## 2.49.8 List.DataElementOutput

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.DataElementOutput** element specifies whether a [List](#) is included in a **data rendering**. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is one of the following:

**Auto** (default): Specifies the default setting for how the list will be included in a data rendering.

**Output**: Specifies that the list is included in a data rendering.

**NoOutput**: Specifies that the list is not included in a data rendering.

**ContentsOnly**: Specifies that the list is included in a data rendering.

The value of the **List.DataElementOutput** element is interpreted as "NoOutput" if all the following conditions are true:

- [List.Visibility](#) is specified for the list.
- [Visibility.Hidden](#) is specified for the **List.Visibility** element.
- [Visibility.ToggleItem](#) is not specified for the **List.Visibility** element.
- The value of **Visibility.Hidden** of the **List.Visibility** element is the **String** literal value "true".

Alternatively, the value of the **List.DataElementOutput** element is interpreted as "NoOutput" if all the following conditions are true:

- The list is contained within a [TableColumn](#), and [TableColumn.Visibility](#) is specified for the **table** column.
- **Visibility.Hidden** is specified for the **TableColumn.Visibility** element.
- **Visibility.ToggleItem** is not specified for the **TableColumn.Visibility** element.
- The value of **Visibility.Hidden** for the **TableColumn.Visibility** element is the **String** literal value "true".

Otherwise, the value of the **List.DataElementOutput** element is interpreted as "Output".

If the **List.DataElementOutput** element is not specified, its value is interpreted as "Auto". If the value of [List.DataInstanceElementOutput](#) is "NoOutput", the list is not included in a data rendering.

Following is the parent element of the **List.DataElementOutput** element.

Parent elements
List

The following is the XML Schema definition of the **List.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

</xsd:element>

## 2.49.9 List.Label

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.Label** element specifies a **document map** label to identify a [List](#) within the rendered [Report](#). This element is optional.

If the **List.Label** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **Variant**. If the expression returns NULL, no item is added to the document map. This element is ignored if the list is contained within a page header or page footer.

Following is the parent element of the **List.Label** element.

Parent elements
List

The following is the XML Schema definition of the **List.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

## 2.49.10 List.Height

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.Height** element specifies the height of a [List](#). This element is optional.

If the **List.Height** element is present, its value MUST be a non-negative [RdlSize](#). If this element is not present, its value is interpreted as the height of the list's container minus the value of the [List.Top](#) element of this list, if specified.

Following is the parent element of the **List.Height** element.

Parent elements
List

The following is the XML Schema definition of the **List.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="0" />
```

## 2.49.11 List.Left

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.Left** element specifies the distance of a [List](#) from the left of the list's container. This element is optional. If this element is present, its value MUST be a non-negative [RdlSize](#) constant. If the **List.Left** element is not present, its value is interpreted as 0.

Following is the parent element of the **List.Left** element.

Parent elements
List

The following is the XML Schema definition of the **List.Left** element.

```
<xsd:element name="Left" type="SizeType" minOccurs="0" />
```

### 2.49.12 List.RepeatWith

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.RepeatWith** element MUST NOT be specified.

Following is the parent element of the **List.RepeatWith** element.

Parent elements
<a href="#">List</a>

The following is the XML Schema definition of the **List.RepeatWith** element.

```
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
```

### 2.49.13 List.ToolTip

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.ToolTip** element specifies a tooltip for a [List](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **List.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not specified, its value is interpreted as NULL.

Following is the parent element of the **List.ToolTip** element.

Parent elements
List

The following is the XML Schema definition of the **List.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

### 2.49.14 List.Top

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.Top** element specifies the distance of a [List](#) from the top of the list's container. This element is optional. If this element is present, its value MUST be a non-negative [RdlSize](#) constant. If this element is not present, the value of the element is interpreted as 0.



Following is the parent element of the **List.Top** element.

Parent elements
List

The following is the XML Schema definition of the **List.Top** element.

```
<xsd:element name="Top" type="SizeType" minOccurs="0" />
```

### 2.49.15 List.Visibility

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.Visibility** element specifies the presentational presence of a [List](#). This element is optional and is of type [Visibility](#).

Following is the parent element of the **List.Visibility** element.

Parent elements
List

The following is the XML Schema definition of the **List.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

### 2.49.16 List.Width

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.Width** element specifies the width of a [List](#). This element is optional. If this element is present, its value MUST be a non-negative [RdlSize](#) constant. If this element is not present, its value is interpreted as the width of the list's container minus the value of the sibling [List.Left](#) element of this list, if specified.

Following is the parent element of the **List.Width** element.

Parent elements
List

The following is the XML Schema definition of the **List.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="0" />
```

### 2.49.17 List.ZIndex

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.ZIndex** element specifies the **ZIndex** of a [List](#). This element is optional. If this element is present, its value MUST be an **UnsignedInt**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **List.ZIndex** element.

Parent elements
List

The following is the XML Schema definition of the **List.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.49.18 List.KeepTogether

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.KeepTogether** element specifies whether the entire contents of a [List](#), including all repeated sections, are to be kept together on one [Page](#) if possible. This element is optional. The value of this element **MUST** be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not specified, its value is interpreted as false.

Following is the parent element of the **List.KeepTogether** element.

Parent elements
List

The following is the XML Schema definition of the **List.KeepTogether** element.

```
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
```

### 2.49.19 List.NoRows

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.NoRows** element specifies text to render instead of the list layout when no rows of data are available for a [List](#). This element is optional. If this element is present, its value **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **List.NoRows** element.

Parent elements
List

The following is the XML Schema definition of the **List.NoRows** element.

```
<xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
```

### 2.49.20 List.PageBreakAtEnd

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.PageBreakAtEnd** element specifies that a renderer inserts a page break at the end of a [List](#). This element is optional. The value of **List.PageBreakAtEnd** **MUST** be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not specified, its value is interpreted as false.

Following is the parent element of the **List.PageBreakAtEnd** element.

Parent elements
List

The following is the XML Schema definition of the **List.PageBreakAtEnd** element.

```
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
```

### 2.49.21 List.PageBreakAtStart

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **List.PageBreakAtStart** element specifies that the renderer inserts a page break at the start of a [List](#). This element is optional. The value of the **List.PageBreakAtStart** element MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not specified, its value is interpreted as false.

Following is the parent element of the **List.PageBreakAtStart** element.

Parent elements
List

The following is the XML Schema definition of the **List.PageBreakAtStart** element.

```
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
```

### 2.49.22 List.DataSetName

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **List.DataSetName** element specifies the name of the **dataset** to use to bind data to a [List](#). This element is optional. If this element is specified, its value MUST be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#), which MUST be the value of the **Name** attribute of a [DataSet](#) that is contained within a [Report](#).

This element MUST be specified if the following conditions are true:

- The list is not contained within another data region.
- More than one **DataSet** element is specified for the report.

This element is ignored for a list that is contained within another data region. A list MUST NOT be specified in a **Report** if no **DataSet** element is specified in the containing report.

If the **List.DataSetName** element is not specified, its value is interpreted as the value of the **DataSetName** of the first ancestor data region in which the list is contained or as the name of the single **DataSet** element that is specified for the report.

Following is the parent element of the **List.DataSetName** element.

Parent elements
List

The following is the XML Schema definition of the **List.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
```

### 2.49.23 List.Filters

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.Filters** element specifies a collection of [Filter](#) elements to be applied to the data for each row of a [List](#). This element is optional and is of type [Filters](#).

Following is the parent element of the **List.Filters** element.

Parent elements
List

The following is the XML Schema definition of the **List.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

### 2.49.24 List.DataInstanceElementOutput

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.DataInstanceElementOutput** element specifies whether the data in the instances of a list is included in a **data rendering** of the [List](#). This element is optional. If specified, the value of the **List.DataInstanceElementOutput** element MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1). The specified value for this element MUST be one of the following:

**Output** (default): Specifies that the list data is included in a data rendering.

**NoOutput**: Specifies that the list data is not included in a data rendering.

**ContentsOnly**: Specifies that the list data is included in a data rendering.

If the **List.DataInstanceElementOutput** element is not specified, its value is interpreted as "Output".

If a [Grouping](#) is specified for the list, the **List.DataInstanceElementOutput** element is ignored, and the value of [Grouping.DataElementOutput](#) MUST be used to determine whether the data in the lists instances is included in a data rendering.

If the data in the list instances is not included in the data rendering, the parent list is not included in the data rendering.

Following is the parent element of the **List.DataInstanceElementOutput** element.

Parent elements
List

The following is the XML Schema definition of the **List.DataInstanceElementOutput** element.

```

<xsd:element name="DataInstanceElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

```

## 2.49.25 List.DataInstanceName

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.DataInstanceName** element specifies the name to use for the data element of each instance of this [List](#) in a **data rendering**. This element is optional. If this element is present, its value MUST be a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as "Item".

If a [Grouping](#) is specified for the list, the **List.DataInstanceName** element is ignored and the value of [Grouping.DataElementName](#) MUST be used for the name of the data element instead.

Following is the parent element of the **List.DataInstanceName** element.

Parent elements
List

The following is the XML Schema definition of the **List.DataInstanceName** element.

```

<xsd:element name="DataInstanceName" type="xsd:string" minOccurs="0" />

```

## 2.49.26 List.FillPage

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.FillPage** element is ignored.

Following is the parent element of the **List.FillPage** element.

Parent elements
<a href="#">List</a>

The following is the XML Schema definition of the **List.FillPage** element.

```

<xsd:element name="FillPage" type="xsd:boolean" minOccurs="0" />

```

## 2.49.27 List.Grouping

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.Grouping** element specifies the expressions by which to group the data in a [List](#). This element is optional and is of type [Grouping](#). This element MUST be specified if there are any data regions contained within the list.

Following is the parent element of the **List.Grouping** element.

Parent elements
<a href="#">List</a>

The following is the XML Schema definition of the **List.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" minOccurs="0" />
```

### 2.49.28 List.ReportItems

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.ReportItems** element specifies a collection of the elements of a list layout. This element is optional and is of type [ReportItems](#).

Following is the parent element of the **List.ReportItems** element.

Parent elements
<a href="#">List</a>

The following is the XML Schema definition of the **List.ReportItems** element.

```
<xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
```

### 2.49.29 List.Sorting

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **List.Sorting** element specifies the expressions by which to sort repeated list regions. This element is optional and is of type [Sorting](#).

Following is the parent element of the **List.Sorting** element.

Parent elements
<a href="#">List</a>

The following is the XML Schema definition of the **List.Sorting** element.

```
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
```

## 2.50 Matrix

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Matrix** element specifies a grid of regions that repeats with each column group and row group in the **rowset**. The grid consists of columns and rows that can optionally be filtered, sorted, grouped, nested, and repeated. **Matrices** provide functionality similar to crosstabs and pivot tables. Unlike a [Table](#), which has a static set of columns, matrix columns can be dynamic.

A matrix has a number of components; these are the corner, **dynamic row** and **static row** group headers, **dynamic column** and **static column** group headers, and detail cells. The corner is a single cell in the upper-left corner. The corner cell can be used to display a label for the matrix or can be left empty. When the **report** is rendered, dynamic **column headers** expand right or left for as many columns as there are groups. Dynamic **row headers** expand down the **page**. The data that appears in the detail cells are aggregates based on the intersections of columns and rows.

The following are the parent elements, attributes, and child elements of the **Matrix** element.

Parent elements
<a href="#">ReportItems</a>
<a href="#">CustomReportItem.AltReportItem</a>

Attributes
<a href="#">Matrix.Name</a>

Child elements
<a href="#">Matrix.Style</a>
<a href="#">Matrix.Action</a>
<a href="#">Matrix.LinkToChild</a>
<a href="#">Matrix.Bookmark</a>
<a href="#">Matrix.CustomProperties</a>
<a href="#">Matrix.DataElementName</a>
<a href="#">Matrix.DataElementOutput</a>
<a href="#">Matrix.Label</a>
<a href="#">Matrix.Height</a>
<a href="#">Matrix.Left</a>
<a href="#">Matrix.RepeatWith</a>
<a href="#">Matrix.ToolTip</a>
<a href="#">Matrix.Top</a>
<a href="#">Matrix.Visibility</a>
<a href="#">Matrix.Width</a>
<a href="#">Matrix.ZIndex</a>

Child elements
<a href="#">Matrix.KeepTogether</a>
<a href="#">Matrix.NoRows</a>
<a href="#">Matrix.PageBreakAtEnd</a>
<a href="#">Matrix.PageBreakAtStart</a>
<a href="#">Matrix.DataSetName</a>
<a href="#">Matrix.Filters</a>
<a href="#">Matrix.CellDataElementName</a>
<a href="#">Matrix.CellDataElementOutput</a>
<a href="#">Matrix.ColumnGroupings</a>
<a href="#">Matrix.Corner</a>
<a href="#">Matrix.GroupsBeforeRowHeaders</a>
<a href="#">Matrix.LayoutDirection</a>
<a href="#">Matrix.MatrixColumns</a>
<a href="#">Matrix.MatrixRows</a>
<a href="#">Matrix.RowGroupings</a>

The following is the XML Schema definition of the **Matrix** element.

```

<xsd:complexType name="MatrixType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
    <xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="Corner" type="CornerType" minOccurs="0" />
    <xsd:element name="ColumnGroupings" type="ColumnGroupingsType" />
    <xsd:element name="RowGroupings" type="RowGroupingsType" />
    <xsd:element name="MatrixRows" type="MatrixRowsType" />
    <xsd:element name="MatrixColumns" type="MatrixColumnsType" />
    <xsd:element name="LayoutDirection" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="LTR" />
          <xsd:enumeration value="RTL" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>

```



```

        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="GroupsBeforeRowHeaders" type="xsd:unsignedInt"
    minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Output" />
            <xsd:enumeration value="NoOutput" />
            <xsd:enumeration value="ContentsOnly" />
            <xsd:enumeration value="Auto" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="CellDataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="CellDataElementOutput" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Output" />
            <xsd:enumeration value="NoOutput" />
            <xsd:enumeration value="ContentsOnly" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.50.1 Matrix.Name

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.Name** attribute specifies a unique identifier for a [Matrix](#). This attribute MUST be specified. The value of the **Matrix.Name** attribute MUST be a **NormalizedString** that is a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#). This value MUST be unique among the set of all report item and scope names.

Following is the parent element of the **Matrix.Name** attribute.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.50.2 Matrix.Style

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.Style** element specifies the style information for a [Matrix](#). This element is optional and is of type [Style](#).

Following is the parent element of the **Matrix.Style** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.50.3 Matrix.Action

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Matrix.Action** element is ignored.

Following is the parent element of the **Matrix.Action** element.

Parent elements
<a href="#">Matrix</a>

The following is the XML Schema definition of the **Matrix.Action** element.

```
<xsd:element name="Action" type="ActionType" minOccurs="0" />
```

### 2.50.4 Matrix.LinkToChild

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Matrix.LinkToChild** element is ignored.

Following is the parent element of the **Matrix.LinkToChild** element.

Parent element
<a href="#">Matrix</a>

The following is the XML Schema definition of the **Matrix.LinkToChild** element.

```
<xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
```

### 2.50.5 Matrix.Bookmark

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Matrix.Bookmark** element specifies a bookmark for a [Matrix](#) that can be linked to via a bookmark action. This element is optional. If the **Matrix.Bookmark** element is specified, its value MUST be a [String](#) or be an expression that evaluates to a **String**.

Following is the parent element of the **Matrix.Bookmark** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.Bookmark** element.

```
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
```

## 2.50.6 Matrix.CustomProperties

**Applies to** [RDL 2005/01](#)

The **Matrix.CustomProperties** element specifies extended information in a collection of name/value pairs. This element is optional and is of type [CustomProperties](#).

Following is the parent element of the **Matrix.CustomProperties** element.

Parent elements
<a href="#">Matrix</a>

The following is the XML Schema definition of the **Matrix.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

In [RDL 2003/10](#), the equivalent element of **Matrix.CustomProperties** is **Matrix.Custom**, which is of type [Custom](#).

## 2.50.7 Matrix.DataElementName

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Matrix.DataElementName** element specifies the name to use for the **data element** of the [Matrix](#) in a **data rendering**.

This element is optional. If this element is not present, its value MUST be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted to be the value of the Name attribute of the **Matrix**.

Following is the parent element of the **Matrix.DataElementName** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

## 2.50.8 Matrix.DataElementOutput

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Matrix.DataElementOutput** element specifies whether a [Matrix](#) is included in a **data rendering**. This element is optional. If the **Matrix.DataElementOutput** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**Auto** (default): Specifies the default setting for how a **matrix** will be included in a data rendering.

**Output:** Specifies that the matrix is included in a data rendering.

**NoOutput:** Specifies that the matrix is not included in a data rendering.

**ContentsOnly:** Specifies that the matrix is included in a data rendering.

The value of the **Matrix.DataElementOutput** is interpreted as "NoOutput" if all the following conditions are true:

- [Matrix.Visibility](#) is specified for the matrix.
- [Visibility.Hidden](#) is specified for the **Matrix.Visibility** element.
- [Visibility.ToggleItem](#) for the **Matrix.Visibility** element is not specified.
- The value of **Visibility.Hidden** for the **Matrix.Visibility** element is the **String** literal value "true".

Alternatively, the value of the **Matrix.DataElementOutput** is interpreted as "NoOutput" if all the following conditions are true:

- The matrix is contained within a table column.
- [TableColumn.Visibility](#) is specified for the [TableColumn](#).
- **Visibility.Hidden** is specified for the **TableColumn.Visibility** element.
- **Visibility.ToggleItem** for the **TableColumn.Visibility** element is not specified.
- The value of **Visibility.Hidden** for the **TableColumn.Visibility** element is the **String** literal value "true".

Otherwise, the value of the **Matrix.DataElementOutput** element is interpreted as "Output".

If the **Matrix.DataElementOutput** element is not specified, its value is interpreted as "Auto".

Following is the parent element of the **Matrix.DataElementOutput** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.50.9 Matrix.Label

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Matrix.Label** element specifies a document map label to identify a [Matrix](#) within the rendered [Report](#).

This element is optional. If **Matrix.Label** is specified, its value MUST be a [String](#) or an expression that evaluates to a **Variant**. If the expression returns NULL, no item is added to the document map. This element is ignored if the **matrix** is contained within a page header or page footer.

Following is the parent element of the **Matrix.Label** element.

Parent elements
<a href="#">Matrix</a>

The following is the XML Schema definition of the **Matrix.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

### 2.50.10 Matrix.Height

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Matrix.Height** element specifies the height of a [Matrix](#). This element is optional. If this element is specified, its value MUST be a non-negative [RDLSize](#) constant.

If specified, the value of **Matrix.Height** after performing the above validations will always be ignored. Regardless of whether the element is specified, the value of **Matrix.Height** MUST be derived from the sum of the heights of the component parts of the matrix, which are the matrix detail rows and column headers.

Following is the parent element of the **Matrix.Height** element.

Parent elements
<a href="#">Matrix</a>

The following is the XML Schema definition of the **Matrix.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="0" />
```

### 2.50.11 Matrix.Left

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Matrix.Left** element specifies the distance of a matrix from the left of the matrix's container. This element is optional. If this element is present, its value MUST be a non-negative [RDLSize](#) constant. If the **Matrix.Left** element is not present, its value is interpreted as "0".

Following is the parent element of the **Matrix.Left** element.

Parent elements
<a href="#">Matrix</a>

The following is the XML Schema definition of the **Matrix.Left** element.

```
<xsd:element name="Left" type="SizeType" minOccurs="0" />
```

### 2.50.12 Matrix.RepeatWith

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.RepeatWith** element MUST NOT be specified.

Following is the parent element of the **Matrix.RepeatWith** element.

Parent elements
<a href="#">Matrix</a>

The following is the XML Schema definition of the **Matrix.RepeatWith** element.

```
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
```

### 2.50.13 Matrix.ToolTip

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.ToolTip** element specifies a tooltip for a [Matrix](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **Matrix.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as NULL.

Following is the parent element of the **Matrix.ToolTip** element.

Parent elements
<a href="#">Matrix</a>

The following is the XML Schema definition of the **Matrix.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

### 2.50.14 Matrix.Top

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.Top** element specifies the distance of a [Matrix](#) from the top of the matrix's container. This element is optional. If this element is specified, its value MUST be a non-negative [RDLSIZE](#). If this element is not specified, the value of the element is interpreted as 0.

Following is the parent element of the **Matrix.Top** element.

Parent elements
<a href="#">Matrix</a>

The following is the XML Schema definition of the **Matrix.Top** element.

```
<xsd:element name="Top" type="SizeType" minOccurs="0" />
```

### 2.50.15 Matrix.Visibility

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.Visibility** element specifies the presentational presence of a [Matrix](#). This element is optional and is of type [Visibility](#).

Following is the parent element of the **Matrix.Visibility** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

### 2.50.16 Matrix.Width

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.Width** element specifies the width of a [Matrix](#). This element is optional. If this element is specified, its value MUST be a non-negative [RDLSize](#).

If specified, the value of **Matrix.Width** after performing the above validations will always be ignored. Regardless of whether it is specified, the value of **Matrix.Width** is derived from the sum of the widths of the component parts of the matrix that are the matrix details columns and row headers.

Following is the parent element of the **Matrix.Width** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="0" />
```

### 2.50.17 Matrix.ZIndex

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.ZIndex** element specifies the **ZIndex** of a [Matrix](#). This element is optional. If this element is present, its value MUST be an **UnsignedInt**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **Matrix.ZIndex** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.50.18 Matrix.KeepTogether

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.KeepTogether** element specifies whether the entire contents of a [Matrix](#), including all repeated sections, are to be kept together on one [Page](#) if possible. This element is optional. The value of this element MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not specified, its value is interpreted as false.

Following is the parent element of the **Matrix.KeepTogether** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.KeepTogether** element.

```
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
```

### 2.50.19 Matrix.NoRows

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.NoRows** element specifies text to render instead of the matrix layout when no rows of data are available for a [Matrix](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Matrix.NoRows** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.NoRows** element.

```
<xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
```

### 2.50.20 Matrix.PageBreakAtEnd

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.PageBreakAtEnd** element specifies whether a renderer inserts a page break at the end of this [Matrix](#). This element is optional. The value of **Matrix.PageBreakAtEnd** MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Matrix.PageBreakAtEnd** element.

Parent elements
Matrix



The following is the XML Schema definition of the **Matrix.PageBreakAtEnd** element.

```
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
```

### 2.50.21 Matrix.PageBreakAtStart

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.PageBreakAtStart** element specifies whether a renderer inserts a page break at the start of this [Matrix](#). This element is optional. The value of **Matrix.PageBreakAtStart** MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not specified, its value is interpreted as false.

Following is the parent element of the **Matrix.PageBreakAtStart** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.PageBreakAtStart** element.

```
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
```

### 2.50.22 Matrix.DataSetName

The **Matrix.DataSetName** element specifies the name of the [DataSet](#) to use to bind data to a [Matrix](#). This element is optional. If this element is present, its value MUST be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is the value of the **Name** attribute of a **DataSet** contained within the [Report](#).

This element MUST be specified if both of the following statements are true:

- The matrix is not contained within another data region.
- There is more than one **dataset** specified for the report.

This element is ignored for a matrix that is contained within another data region. A matrix MUST NOT be specified in a report if there is no **DataSet** element specified in the containing report. If this element is not specified, its value is interpreted as the value of the **DataSetName** of the first ancestor data region the matrix is contained within or the name of the single **DataSet** element specified for the report.

Following is the parent element of the **Matrix.DataSetName** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
```

### 2.50.23 Matrix.Filters

The **Matrix.Filters** element specifies a collection of [Filter](#) elements to be applied to the data for each row of the [Matrix](#). This element is optional and is of type [Filters](#).

Following is the parent element of the **Matrix.Filters** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

### 2.50.24 Matrix.CellDataElementName

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.CellDataElementName** element specifies the name to use for the **data element** of each instance of the cell that contains the detail data of a [Matrix](#) in a **data rendering**. The **Matrix.CellDataElementName** element is optional. If this element is present, its value MUST be a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as "Cell".

Following is the parent element of the **Matrix.CellDataElementName** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.CellDataElementName** element.

```
<xsd:element name="CellDataElementName" type="xsd:string" minOccurs="0" />
```

### 2.50.25 Matrix.CellDataElementOutput

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.CellDataElementOutput** element specifies whether the data in a details cell in a [Matrix](#) is included in a **data rendering** of the matrix. The **Matrix.CellDataElementOutput** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). The specified value for this element MUST be one of the following:

**Output** (Default): Specifies that the cell detail data is included in a data rendering.

**NoOutput**: Specifies that the cell detail data is not included in a data rendering.

**ContentsOnly**: Specifies that the cell itself is not included in a data rendering, but the contents of the cell are included in the data rendering and are rendered as if the contents were in the cell's container.

If this element is not present, its value is interpreted as "Output".

Following is the parent element of the **Matrix.CellDataElementOutput** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.CellDataElementOutput** element.

```
<xsd:element name="CellDataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.50.26 Matrix.ColumnGroupings

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.ColumnGroupings** element specifies the collection of column **groupings** for a matrix. This element **MUST** be specified. The **Matrix.ColumnGroupings** element is of type [ColumnGroupings](#).

Following is the parent element of the **Matrix.ColumnGroupings** element.

Parent elements
<a href="#">Matrix</a>

The following is the XML Schema definition of the **Matrix.ColumnGroupings** element.

```
<xsd:element name="ColumnGroupings" type="ColumnGroupingsType" />
```

### 2.50.27 Matrix.Corner

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.Corner** element specifies the region that contains the elements of the upper-left corner area of a [Matrix](#). This element is optional and is of type [Corner](#). If this element is not present, no report items are rendered in the corner.

Following is the parent element of the **Matrix.Corner** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.Corner** element.

```
<xsd:element name="Corner" type="CornerType" minOccurs="0" />
```

## 2.50.28 Matrix.GroupsBeforeRowHeaders

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.GroupsBeforeRowHeaders** element specifies the number of instances of the outermost column group to render before the row headers in a [Matrix](#). This element is optional, and its value MUST be an **UnsignedInt**. If this element is not specified, its value is interpreted as 0.

If the [Matrix.LayoutDirection](#) element value is "LTR", this element specifies the number of instances of the first matrix column group to appear to the left of the matrix row headers.

If the **Matrix.LayoutDirection** element value is "RTL", this element specifies the number of instances of the first matrix column group to appear to the right of the matrix row headers.

Following is the parent element of the **Matrix.GroupsBeforeRowHeaders** element.

Parent elements
<a href="#">Matrix</a>

The following is the XML Schema definition of the **Matrix.GroupsBeforeRowHeaders** element.

```
<xsd:element name="GroupsBeforeRowHeaders" type="xsd:unsignedInt" minOccurs="0" />
```

## 2.50.29 Matrix.LayoutDirection

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.LayoutDirection** element specifies whether matrix columns grow left-to-right (with headers on the left) or right-to-left (with headers on the right). This element is optional and if specified is of type [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). The value of the **Matrix.LayoutDirection** element MUST be one of the following:

**LTR** (default): Specifies that the matrix columns are laid out in sequence, starting at the left and progressing to the right.

**RTL**: Specifies that the matrix columns are laid out in sequence, starting at the right and progressing to the left.

If this element is not specified, its value is interpreted as "LTR".

Following is the parent element of the **Matrix.LayoutDirection** element.

Parent elements
<a href="#">Matrix</a>

The following is the XML Schema definition of the **Matrix.LayoutDirection** element.

```
<xsd:element name="LayoutDirection" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="LTR" />
      <xsd:enumeration value="RTL" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.50.30 Matrix.MatrixColumns

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.MatrixColumns** element specifies the collection of columns contained in each detail cell of a matrix layout. This element **MUST** be specified. The **Matrix.MatrixColumns** element is of type [MatrixColumns](#).

Following is the parent element of the **Matrix.MatrixColumns** element.

Parent elements
<a href="#">Matrix</a>

The following is the XML Schema definition of the **Matrix.MatrixColumns** element.

```
<xsd:element name="MatrixColumns" type="MatrixColumnsType" />
```

### 2.50.31 Matrix.MatrixRows

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.MatrixRows** element specifies the collection of rows contained in each detail cell of a matrix layout. This element **MUST** be specified. The **Matrix.MatrixRows** element is of type [MatrixRows](#).

Following is the parent element of the **Matrix.MatrixRows** element.

Parent elements
<a href="#">Matrix</a>

The following is the XML Schema definition of the **Matrix.MatrixRows** element.

```
<xsd:element name="MatrixRows" type="MatrixRowsType" />
```

### 2.50.32 Matrix.RowGroupings

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Matrix.RowGroupings** element specifies the collection of row **groupings** for a [Matrix](#). This element **MUST** be specified. The **Matrix.MatrixRows** element is of type [RowGroupings](#).

Following is the parent element of the **Matrix.MatrixRowGroupings** element.

Parent elements
Matrix

The following is the XML Schema definition of the **Matrix.MatrixRowGroupings** element.

```
<xsd:element name="RowGroupings" type="RowGroupingsType" />
```

## 2.51 ColumnGroupings

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **ColumnGroupings** element specifies the collection of column groupings in a [Matrix](#). If the **ColumnGroupings** element is specified, there MUST be at least one and there can be more than one [ColumnGrouping](#) in the **ColumnGroupings** collection.

The following are the parent and child elements of the **ColumnGroupings** element.

Parent elements
Matrix

Child elements
<a href="#">ColumnGroupings.ColumnGrouping</a>

The following is the XML Schema definition of the **ColumnGroupings** element.

```
<xsd:complexType name="ColumnGroupingsType">
  <xsd:sequence>
    <xsd:element name="ColumnGrouping" type="ColumnGroupingType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.51.1 ColumnGroupings.ColumnGrouping

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **ColumnGroupings.ColumnGrouping** element specifies a member of the collection of column groupings in a [Matrix](#). This element MUST be specified and is of type [ColumnGrouping](#). More than one **ColumnGroupings.ColumnGrouping** element can be specified under a single [ColumnGroupings](#) parent element. Each column grouping specifies a row of column headers. The first column grouping is the outermost row of column headers.

Following is the parent element of the **ColumnGroupings.ColumnGrouping** element.

Parent elements
ColumnGroupings

The following is the XML Schema definition of the **ColumnGroupings.ColumnGrouping** element.

```
<xsd:element name="ColumnGrouping" type="ColumnGroupingType" maxOccurs="unbounded" />
```

## 2.52 ColumnGrouping

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **ColumnGrouping** element specifies a **column header** region for a [Matrix](#). A **ColumnGrouping** element MUST have either the [DynamicColumns](#) or the [StaticColumns](#) child element, but not both.

The following are the parent and child elements of the **ColumnGrouping** element.

Parent elements
<a href="#">ColumnGroupings</a>

Child elements
<a href="#">ColumnGrouping.DynamicColumns</a>
<a href="#">ColumnGrouping.FixedHeader</a>
<a href="#">ColumnGrouping.Height</a>
<a href="#">ColumnGrouping.StaticColumns</a>

The following is the XML Schema definition of the **ColumnGrouping** element.

```
<xsd:complexType name="ColumnGroupingType">  
  <xsd:choice minOccurs="1" maxOccurs="unbounded">  
    <xsd:element name="Height" type="SizeType" />  
    <xsd:element name="FixedHeader" type="xsd:boolean" minOccurs="0" />  
    <xsd:element name="DynamicColumns" type="DynamicColumnsRowsType" minOccurs="0" />  
    <xsd:element name="StaticColumns" type="StaticColumnsType" minOccurs="0" />  
    <xsd:any namespace="##other" processContents="skip" />  
  </xsd:choice>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

### 2.52.1 ColumnGrouping.DynamicColumns

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ColumnGrouping.DynamicColumns** element specifies the **dynamic column** headers for a column grouping in a [Matrix](#). This element is optional and is of type [DynamicColumns](#).

If **ColumnGrouping.DynamicColumns** is not specified, [ColumnGrouping.StaticColumns](#) MUST be specified. If **ColumnGrouping.DynamicColumns** is specified, **ColumnGrouping.StaticColumns** MUST NOT be specified.

Following is the parent element of the **ColumnGrouping.DynamicColumns** element.

Parent elements
<a href="#">ColumnGrouping</a>

The following is the XML Schema definition of the **ColumnGrouping.DynamicColumns** element.

```
<xsd:element name="DynamicColumns" type="DynamicColumnsRowsType" minOccurs="0" />
```

### 2.52.2 ColumnGrouping.FixedHeader

**Applies to** [RDL 2005/01](#)

The **ColumnGrouping.FixedHeader** element specifies whether the header for a column grouping remains visible when a [Matrix](#) is partially scrolled off the [Page](#). Fixed headers MUST be contiguous and MUST include the outermost grouping.

This element is optional. The value of this element MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **ColumnGrouping.FixedHeader** element.

Parent elements
<a href="#">ColumnGrouping</a>

The following is the XML Schema definition of the **ColumnGrouping.FixedHeader** element.

```
<xsd:element name="FixedHeader" type="xsd:boolean" minOccurs="0" />
```

### 2.52.3 ColumnGrouping.Height

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **ColumnGrouping.Height** element specifies the height of a column header in a [Matrix](#). This element MUST be specified. The value of this element MUST be a non-negative [RdlSize](#).

Following is the parent element of the **ColumnGrouping.Height** element.

Parent elements
<a href="#">ColumnGrouping</a>

The following is the XML Schema definition of the **ColumnGrouping.Height** element.

```
<xsd:element name="Height" type="SizeType" />
```

### 2.52.4 ColumnGrouping.StaticColumns

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **ColumnGrouping.StaticColumns** element specifies the **static column** headers for a column grouping in a [Matrix](#). This element is optional and is of type [StaticColumns](#).

If the **ColumnGrouping.StaticColumns** element is not specified, [ColumnGrouping.DynamicColumns](#) MUST be specified. If **ColumnGrouping.StaticColumns** is specified, **ColumnGrouping.DynamicColumns** MUST NOT be specified.

Following is the parent element of the **ColumnGrouping.StaticColumns** element.

Parent elements
<a href="#">ColumnGrouping</a>

The following is the XML Schema definition of the **ColumnGrouping.StaticColumns** element.

```
<xsd:element name="StaticColumns" type="StaticColumnsType" minOccurs="0" />
```



## 2.53 DynamicColumns

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicColumns** element specifies a **column header** region that repeats with each column group in a **rowset** in a [Matrix](#).

The following are the parent and child elements of the **DynamicColumns** element.

Parent elements
<a href="#">ColumnGrouping</a>

Child elements
<a href="#">DynamicColumns.Grouping</a>
<a href="#">DynamicColumns.ReportItems</a>
<a href="#">DynamicColumns.Sorting</a>
<a href="#">DynamicColumns.Subtotal</a>
<a href="#">DynamicColumns.Visibility</a>

The following is the XML Schema definition of the **DynamicColumns** element.

```
<xsd:complexType name="DynamicColumnsRowsType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Grouping" type="GroupingType" />
    <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
    <xsd:element name="Subtotal" type="SubtotalType" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.53.1 DynamicColumns.Grouping

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicColumns.Grouping** element specifies the expressions by which to group the data in a column group in a [Matrix](#). This element MUST be specified and is of type [Grouping](#).

Following is the parent element of the **DynamicColumns.Grouping** element.

Parent elements
<a href="#">DynamicColumns</a>

The following is the XML Schema definition of the **DynamicColumns.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" />
```

## 2.53.2 DynamicColumns.ReportItems

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicColumns.ReportItems** element specifies a collection of the elements of the column header layout in a [Matrix](#). This element MUST be specified and is of type [ReportItems](#). The **ReportItems** collection MUST contain exactly one report item. The **Top**, **Left**, **Height**, and **Width** elements for this report item are ignored, and their values are interpreted as follows:

- Position: "0, 0"
- Size: "100%, 100%"

Following is the parent element of the **DynamicColumns.ReportItems** element.

Parent elements
<a href="#">DynamicColumns</a>

The following is the XML Schema definition of the **DynamicColumns.ReportItems** element.

```
<xsd:element name="ReportItems" type="ReportItemsType" />
```

## 2.53.3 DynamicColumns.Sorting

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicColumns.Sorting** element specifies the expressions by which to sort the columns in a column group in a [Matrix](#). This element is optional and is of type [Sorting](#).

Following is the parent element of the **DynamicColumns.Sorting** element.

Parent elements
<a href="#">DynamicColumns</a>

The following is the XML Schema definition of the **DynamicColumns.Sorting** element.

```
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
```

## 2.53.4 DynamicColumns.Subtotal

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicColumns.Subtotal** element specifies that an automatic subtotal column is included in the rendering of a [Matrix](#). This element is optional and is of type [Subtotal](#).

Following is the parent element of the **DynamicColumns.Subtotal** element.

Parent elements
<a href="#">DynamicColumns</a>

The following is the XML Schema definition of the **DynamicColumns.Subtotal** element.

```
<xsd:element name="Subtotal" type="SubtotalType" minOccurs="0" />
```

### 2.53.5 DynamicColumns.Visibility

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicColumns.Visibility** element specifies the presentational presence of the **dynamic columns** for a grouping in a [Matrix](#). This element is optional and is of type [Visibility](#).

Following is the parent element of the **DynamicColumns.Visibility** element.

Parent elements
<a href="#">DynamicColumns</a>

The following is the XML Schema definition of the **DynamicColumns.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

### 2.54 Subtotal

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Subtotal** element specifies a subtotal column or row to include in a [Matrix](#). The subtotal is specified for a [DynamicColumns](#) element or a [DynamicRows](#) element. The subtotal is applied to the full set of data for the entire row or column group that is specified by the parent element.

The following are the parent and child elements of the **Subtotal** element.

Parent elements
<a href="#">DynamicColumns</a>
<a href="#">DynamicRows</a>

Child elements
<a href="#">Subtotal.DataElementName</a>
<a href="#">Subtotal.DataElementOutput</a>
<a href="#">Subtotal.Position</a>
<a href="#">Subtotal.ReportItems</a>
<a href="#">Subtotal.Style</a>

The following is the XML Schema definition of the **Subtotal** element.

```
<xsd:complexType name="SubtotalType">  
  <xsd:choice minOccurs="1" maxOccurs="unbounded">  
    <xsd:element name="ReportItems" type="ReportItemsType" />  
    <xsd:element name="Style" type="StyleType" minOccurs="0" />  
    <xsd:element name="Position" minOccurs="0">  
      <xsd:simpleType>
```

```

    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Before" />
      <xsd:enumeration value="After" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.54.1 Subtotal.DataElementName

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Subtotal.DataElementName** element specifies the name to use for the **data element** of the [Subtotal](#) in a **data rendering**. This element is optional. If this element is present, its value MUST be a case-sensitive **CLS-compliant identifier** [UTR15]. If this element is not present, its value is interpreted as a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that has the value "Total".

Following is the parent element of the **Subtotal.DataElementName** element.

Parent elements
Subtotal

The following is the XML Schema definition of the **Subtotal.DataElementName** element.

```

<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />

```

### 2.54.2 Subtotal.DataElementOutput

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Subtotal.DataElementOutput** element specifies whether a [Subtotal](#) is included in a [Matrix](#) in a **data rendering**. This element is optional. If the **Subtotal.DataElementOutput** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1).

The value of this element MUST be one of the following:

**Output** (default): Specifies that the subtotal is included in a data rendering.

**NoOutput**: Specifies that the subtotal is not included in a data rendering.

**ContentsOnly**: Specifies that the subtotal is included in a data rendering.

If this element is not specified, its value is interpreted as "Output".

Following is the parent element of the **Subtotal.DataElementOutput** element.

<b>Parent elements</b>
Subtotal

The following is the XML Schema definition of the **Subtotal.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.54.3 Subtotal.Position

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Subtotal.Position** element specifies whether a [Subtotal](#) column or row is rendered before or after the detail columns or rows in a [Matrix](#). A subtotal column is rendered before or after a detail column if it is rendered to the left or right of the detail column, respectively. A subtotal row is rendered before or after a detail row if it is rendered above or below the detail row, respectively.

This element is optional. If this element is specified, it is of type [String](#). The value of the **Subtotal.Position** element MUST be one of the following:

**Before:** Specifies that the subtotal column or row is rendered before the detail columns or rows. "Before" means to the left of detail columns or above detail rows.

**After** (default): Specifies that the subtotal column or row is rendered after the detail columns or rows. "After" means to the right of detail columns or below detail rows.

If this element is not present, its value is interpreted as "After".

Following is the parent element of the **Subtotal.Position** element.

<b>Parent elements</b>
Subtotal

The following is the XML Schema definition of the **Subtotal.Position** element.

```
<xsd:element name="Position" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Before" />
      <xsd:enumeration value="After" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.54.4 Subtotal.ReportItems

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Subtotal.ReportItems** element specifies the header cell for a [Subtotal](#) column or row in a [Matrix](#). This element MUST be specified and is of type [ReportItems](#). This **ReportItems** collection MUST contain exactly one **report item**, which MUST be a [Textbox](#). The **Top**, **Left**, **Height**, and **Width** elements for this report item are ignored, and their values are interpreted as follows:

- Position: "0, 0"
- Size: "100%, 100%"

Following is the parent element of the **Subtotal.ReportItems** element.

Parent elements
Subtotal

The following is the XML Schema definition of the **Subtotal.ReportItems** element.

```
<xsd:element name="ReportItems" type="ReportItemsType" />
```

### 2.54.5 Subtotal.Style

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Subtotal.Style** element specifies the style information for a [Subtotal](#) column or row in a [Matrix](#). This element is optional and is of type [Style](#).

The subtotal style properties override the style properties for all top-level report items that are contained in the matrix body cells of a subtotal column or row. At the intersections of subtotal columns and subtotal rows, the subtotal row style takes priority.

Following is the parent element of the **Subtotal.Style** element.

Parent elements
Subtotal

The following is the XML Schema definition of the **Subtotal.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.55 StaticColumns

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **StaticColumns** element specifies a collection of **column header** regions with a fixed collection of columns in a [Matrix](#). Only one [ColumnGrouping](#) in each matrix can be static. If the **StaticColumns** element is specified, there MUST be at least one and there can be more than one [StaticColumn](#) in the **StaticColumns** collection.

The following are the parent and child elements of the **StaticColumns** element.

Parent elements
ColumnGrouping

<b>Child elements</b>
-----------------------

<a href="#">StaticColumns.StaticColumn</a>
--

The following is the XML Schema definition of the **StaticColumns** element.

```
<xsd:complexType name="StaticColumnsType">
  <xsd:sequence>
    <xsd:element name="StaticColumn" type="StaticColumnType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.55.1 StaticColumns.StaticColumn

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **StaticColumns.StaticColumn** element specifies a **static column** header for a column grouping in a [Matrix](#). This element MUST be specified and is of type [StaticColumn](#). More than one **StaticColumns.StaticColumn** element can be specified under a single [StaticColumns](#) parent element.

Following is the parent element of the **StaticColumns.StaticColumn** element.

<b>Parent elements</b>
------------------------

<a href="#">StaticColumns</a>
-------------------------------

The following is the XML Schema definition of the **StaticColumns.StaticColumn** element.

```
<xsd:element name="StaticColumn" type="StaticColumnType" maxOccurs="unbounded" />
```

### 2.56 StaticColumn

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **StaticColumn** element specifies a fixed column header region in a [Matrix](#).

The following are the parent and child elements of the **StaticColumn** element.

<b>Parent elements</b>
------------------------

<a href="#">StaticColumns</a>
-------------------------------

<b>Child elements</b>
-----------------------

<a href="#">StaticColumn.ReportItems</a>
--

The following is the XML Schema definition of the **StaticColumn** element.

```
<xsd:complexType name="StaticColumnType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
```

```

    <xsd:element name="ReportItems" type="ReportItemsType" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.56.1 StaticColumn.ReportItems

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **StaticColumn.ReportItems** element specifies the collection of the report items of the column header layout in a [Matrix](#). This element MUST be specified and is of type [ReportItems](#). The **ReportItems** collection MUST contain exactly one report item. The **Top**, **Left**, **Height**, and **Width** elements for this report item are ignored, and their values are interpreted as follows:

- Position: "0, 0"
- Size: "100%, 100%"

Following is the parent element of the **StaticColumn.ReportItems** element.

Parent elements
<a href="#">StaticColumn</a>

The following is the XML Schema definition of the **StaticColumn.ReportItems** element.

```

<xsd:element name="ReportItems" type="ReportItemsType" />

```

### 2.57 Corner

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **Corner** element specifies the layout and structure of the upper-left corner region of a [Matrix](#).

The following are the parent and child elements of the **Corner** element.

Parent elements
<a href="#">Matrix</a>

Child elements
<a href="#">Corner.ReportItems</a>

The following is the XML Schema definition of the **Corner** element.

```

<xsd:complexType name="CornerType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```



```
</xsd:complexType>
```

### 2.57.1 Corner.ReportItems

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **Corner.ReportItems** element specifies the region that contains the elements of the corner layout in a [Matrix](#). This element **MUST** be specified and is of type [ReportItems](#). This **ReportItems** collection **MUST** contain exactly one **report item**. The **Top**, **Left**, **Height**, and **Width** elements for this report item are ignored, and their values are interpreted as follows:

- Position: "0, 0"
- Size: "100%, 100%"

Following is the parent element of the **Corner.ReportItems** element.

Parent elements
<a href="#">Corner</a>

The following is the XML Schema definition of the **Corner.ReportItems** element.

```
<xsd:element name="ReportItems" type="ReportItemsType" />
```

### 2.58 MatrixColumns

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **MatrixColumns** element specifies the collection of columns in the detail section of a [Matrix](#). If the **MatrixColumns** element is present, there **MUST** be at least one and there can be more than one [MatrixColumn](#) in the **MatrixColumns** collection.

The following are the parent and child elements of the **MatrixColumns** element.

Parent elements
Matrix

Child elements
<a href="#">MatrixColumns.MatrixColumn</a>

The following is the XML Schema definition of the **MatrixColumns** element.

```
<xsd:complexType name="MatrixColumnsType">  
  <xsd:sequence>  
    <xsd:element name="MatrixColumn" type="MatrixColumnType"  
      maxOccurs="unbounded" />  
  </xsd:sequence>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

## 2.58.1 MatrixColumns.MatrixColumn

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **MatrixColumns.MatrixColumn** element specifies a single column in the detail section of a [Matrix](#). This element MUST be specified and is of type [MatrixColumn](#). More than one **MatrixColumns.MatrixColumn** element can be specified under a single [MatrixColumns](#) parent element. There MUST be one **MatrixColumn** per [StaticColumn](#) in the matrix. If there are no static columns, there MUST be exactly one **MatrixColumn**.

Following is the parent element of the **MatrixColumns.MatrixColumn** element.

Parent elements
<a href="#">MatrixColumns</a>

The following is the XML Schema definition of the **MatrixColumns.MatrixColumn** element.

```
<xsd:element name="MatrixColumn" type="MatrixColumnType" maxOccurs="unbounded" />
```

## 2.59 MatrixColumn

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **MatrixColumn** element specifies a column in the detail section of a [Matrix](#).

The following are the parent and child elements of the **MatrixColumn** element.

Parent elements
<a href="#">MatrixColumns</a>

Child elements
<a href="#">MatrixColumn.Width</a>

The following is the XML Schema definition of the **MatrixColumn** element.

```
<xsd:complexType name="MatrixColumnType">  
  <xsd:choice minOccurs="1" maxOccurs="unbounded">  
    <xsd:element name="Width" type="SizeType" />  
    <xsd:any namespace="##other" processContents="skip" />  
  </xsd:choice>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

### 2.59.1 MatrixColumn.Width

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **MatrixColumn.Width** element specifies the width of each detail cell in a column. This element MUST be specified. The value of this element MUST be a non-negative [RdlSize](#).

Following is the parent element of the **MatrixColumn.Width** element.

Parent elements
-----------------

<a href="#">MatrixColumn</a>
------------------------------

The following is the XML Schema definition of the **MatrixColumn.Width** element.

```
<xsd:element name="Width" type="SizeType" />
```

## 2.60 MatrixRows

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **MatrixRows** element specifies the collection of rows in the detail section of a [Matrix](#). If the **MatrixRows** element is specified, there MUST be at least one and there can be more than one [MatrixRow](#) in the **MatrixRows** collection.

The following are the parent and child elements of the **MatrixRows** element.

Parent elements
-----------------

Matrix
--------

Child elements
----------------

<a href="#">MatrixRows.MatrixRow</a>
--------------------------------------

The following is the XML Schema definition of the **MatrixRows** element.

```
<xsd:complexType name="MatrixRowsType">
  <xsd:sequence>
    <xsd:element name="MatrixRow" type="MatrixRowType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.60.1 MatrixRows.MatrixRow

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **MatrixRows.MatrixRow** element specifies a row in the detail section of a [Matrix](#). This element MUST be specified and is of type [MatrixRow](#). More than one **MatrixRows.MatrixRow** element can be specified under a single [MatrixRows](#) parent element. There MUST be one **MatrixRow** per [StaticRow](#) in the matrix. If there are no **static rows**, there MUST be exactly one **MatrixRow**.

Following is the parent element of the **MatrixRows.MatrixRow** element.

Parent elements
-----------------

MatrixRows
------------

The following is the XML Schema definition of the **MatrixRows.MatrixRow** element.

```
<xsd:element name="MatrixRow" type="MatrixRowType" maxOccurs="unbounded" />
```

## 2.61 MatrixRow

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **MatrixRow** element specifies the collection of cells in a row of the detail section of a [Matrix](#).

The following are the parent and child elements of the **MatrixRow** element.

Parent elements
<a href="#">MatrixRows</a>

Child elements
<a href="#">MatrixRow.Height</a>
<a href="#">MatrixRow.MatrixCells</a>

The following is the XML Schema definition of the **MatrixRow** element.

```
<xsd:complexType name="MatrixRowType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="MatrixCells" type="MatrixCellsType" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.61.1 MatrixRow.Height

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **MatrixRow.Height** element specifies the height of each detail cell in a row in a [Matrix](#). This element **MUST** be specified. The value of this element **MUST** be a non-negative [RdlSize](#).

Following is the parent element of the **MatrixRow.Height** element.

Parent elements
<a href="#">MatrixRow</a>

The following is the XML Schema definition of the **MatrixRow.Height** element.

```
<xsd:element name="Height" type="SizeType" />
```

### 2.61.2 MatrixRow.MatrixCells

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **MatrixRow.MatrixCells** element specifies the collection of cells in a row in the detail section of a [Matrix](#). This element **MUST** be specified and is of type [MatrixCells](#).

Following is the parent element of the **MatrixRow.MatrixCells** element.

Parent elements
<a href="#">MatrixRow</a>

The following is the XML Schema definition of the **MatrixRow.MatrixCells** element.

```
<xsd:element name="MatrixCells" type="MatrixCellsType" />
```

## 2.62 MatrixCells

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **MatrixCells** element specifies the set of cells in a row of the detail section of a [Matrix](#). If the **MatrixCells** element is specified, there MUST be at least one and there can be more than one [MatrixCell](#) in the **MatrixCells** collection.

The following are the parent and child elements of the **MatrixCells** element.

Parent elements
<a href="#">MatrixRow</a>

Child elements
<a href="#">MatrixCells.MatrixCell</a>

The following is the XML Schema definition of the **MatrixCells** element.

```
<xsd:complexType name="MatrixCellsType">  
  <xsd:sequence>  
    <xsd:element name="MatrixCell" type="MatrixCellType" maxOccurs="unbounded" />  
  </xsd:sequence>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

### 2.62.1 MatrixCells.MatrixCell

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **MatrixCells.MatrixCell** element specifies a cell in a row in the detail section of a [Matrix](#). This element MUST be specified and is of type [MatrixCell](#). More than one **MatrixCells.MatrixCell** element can be specified under a single [MatrixCells](#) parent element. There MUST be one **MatrixCell** per [StaticColumn](#) in the matrix. If there are no static columns, there MUST be exactly one [MatrixColumn](#).

Following is the parent element of the **MatrixCells.MatrixCell** element.

Parent elements
MatrixCells

The following is the XML Schema definition of the **MatrixCells.MatrixCell** element.

```
<xsd:element name="MatrixCell" type="MatrixCellType" maxOccurs="unbounded" />
```

## 2.63 MatrixCell

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **MatrixCell** element specifies the contents of each detail cell in a [Matrix](#).

Page breaks on report items inside a **MatrixCell** are ignored. For the purposes of [Visibility.ToggleItem](#), a **MatrixCell** is considered to be in the same scope as the matrix. This means that report items contained within a **MatrixCell** do not have their visibility toggled by report items in the matrix row headers or **column headers**.

The following are the parent and child elements of the **MatrixCell** element.

Parent elements
<a href="#">MatrixCells</a>

Child elements
<a href="#">MatrixCell.ReportItems</a>

The following is the XML Schema definition of the **MatrixCell** element.

```
<xsd:complexType name="MatrixCellType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.63.1 MatrixCell.ReportItems

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **MatrixCell.ReportItems** element specifies the collection of **report items** contained in each detail cell of a [Matrix](#) layout. This element MUST be specified and is of type [ReportItems](#). The **ReportItems** collection MUST contain exactly one report item. The **Top**, **Left**, **Height**, and **Width** elements for the report item are ignored, and their values are interpreted as follows:

- Position: "0, 0"
- Size: "100%, 100%"

Following is the parent element of the **MatrixCell.ReportItems** element.

Parent elements
<a href="#">MatrixCell</a>

The following is the XML Schema definition of the **MatrixCell.ReportItems** element.

```
<xsd:element name="ReportItems" type="ReportItemsType" />
```

## 2.64 RowGroupings

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **RowGroupings** element specifies the collection of row groupings for a [Matrix](#). If the **RowGroupings** element is specified, there MUST be at least one and there can be more than one [RowGrouping](#) in the **RowGroupings** collection.

The following are the parent and child elements of the **RowGroupings** element.

Parent elements
Matrix

Child elements
<a href="#">RowGroupings.RowGrouping</a>

The following is the XML Schema definition of the **RowGroupings** element.

```
<xsd:complexType name="RowGroupingsType">
  <xsd:sequence>
    <xsd:element name="RowGrouping" type="RowGroupingType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.64.1 RowGroupings.RowGrouping

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **RowGroupings.RowGrouping** element specifies the collection of row groupings in a [Matrix](#). Each row grouping specifies a column of row headers. The first row grouping is the outermost column of row headers. This element MUST be specified and is of type [RowGrouping](#). More than one **RowGroupings.RowGrouping** element can be specified under a single [RowGroupings](#) parent element.

Following is the parent element of the **RowGroupings.RowGrouping** element.

Parent elements
RowGroupings

The following is the XML Schema definition of the **RowGroupings.RowGrouping** element.

```
<xsd:element name="RowGrouping" type="RowGroupingType" maxOccurs="unbounded" />
```

## 2.65 RowGrouping

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **RowGrouping** element specifies a **row header** region for a [Matrix](#). A **RowGrouping** element MUST have either a [DynamicRows](#) or a [StaticRows](#) child element, but not both.

The following are the parent and child elements of the **RowGrouping** element.

Parent elements
<a href="#">RowGroupings</a>

Child elements
<a href="#">RowGrouping.DynamicRows</a>
<a href="#">RowGrouping.FixedHeader</a>
<a href="#">RowGrouping.StaticRows</a>
<a href="#">RowGrouping.Width</a>

The following is the XML Schema definition of the **RowGrouping** element.

```
<xsd:complexType name="RowGroupingType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Width" type="SizeType" />
    <xsd:element name="FixedHeader" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DynamicRows" type="DynamicColumnsRowsType" minOccurs="0" />
    <xsd:element name="StaticRows" type="StaticRowsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.65.1 RowGrouping.DynamicRows

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **RowGrouping.DynamicRows** element specifies the **dynamic row** headers for this row grouping in a [Matrix](#). This element is optional and, if specified, is of type [DynamicRows](#). If **RowGrouping.DynamicRows** is not specified, [RowGrouping.StaticRows](#) MUST be specified. If **RowGrouping.DynamicRows** is specified, **RowGrouping.StaticRows** MUST NOT be specified.

Following is the parent element of the **RowGrouping.DynamicRows** element.

Parent elements
<a href="#">RowGrouping</a>

The following is the XML Schema definition of the **RowGrouping.DynamicRows** element.

```
<xsd:element name="DynamicRows" type="DynamicColumnsRowsType" minOccurs="0" />
```



## 2.65.2 RowGrouping.FixedHeader

**Applies to** [RDL 2005/01](#)

The **RowGrouping.FixedHeader** element specifies whether the header for this row grouping remains visible when the [Matrix](#) is partially scrolled off the [Page](#). Fixed headers MUST be contiguous and MUST include the outermost grouping.

The **RowGrouping.FixedHeader** element is optional. The value of this element MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **RowGrouping.FixedHeader** element.

Parent elements
<a href="#">RowGrouping</a>

The following is the XML Schema definition of the **RowGrouping.FixedHeader** element.

```
<xsd:element name="FixedHeader" type="xsd:boolean" minOccurs="0" />
```

## 2.65.3 RowGrouping.StaticRows

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **RowGrouping.StaticRows** element specifies the **static row** headers for this row grouping in a [Matrix](#). This element is optional and, if specified, is of type [StaticRows](#). If **RowGrouping.StaticRows** is not specified, [RowGrouping.DynamicRows](#) MUST be specified. If **RowGrouping.StaticRows** is specified, **RowGrouping.DynamicRows** MUST NOT be specified.

Following is the parent element of the **RowGrouping.StaticRows** element.

Parent elements
<a href="#">RowGrouping</a>

The following is the XML Schema definition of the **RowGrouping.StaticRows** element.

```
<xsd:element name="StaticRows" type="StaticRowsType" minOccurs="0" />
```

## 2.65.4 RowGrouping.Width

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **RowGrouping.Width** element specifies the width of a row header in a [Matrix](#). This element MUST be specified. The value of this element MUST be a non-negative [RdlSize](#).

Following is the parent element of the **RowGrouping.Width** element.

Parent elements
<a href="#">RowGrouping</a>

The following is the XML Schema definition of the **RowGrouping.Width** element.

```
<xsd:element name="Width" type="SizeType" />
```

## 2.66 DynamicRows

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicRows** element specifies a **row header** region that repeats with each row group in a **rowset** in a [Matrix](#).

The following are the parent and child elements of the **DynamicRows** element.

Parent elements
<a href="#">RowGrouping</a>

Child elements
<a href="#">DynamicRows.Grouping</a>
<a href="#">DynamicRows.ReportItems</a>
<a href="#">DynamicRows.Sorting</a>
<a href="#">DynamicRows.Subtotal</a>
<a href="#">DynamicRows.Visibility</a>

The following is the XML Schema definition of the **DynamicRows** element.

```
<xsd:complexType name="DynamicColumnsRowsType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Grouping" type="GroupingType" />
    <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
    <xsd:element name="Subtotal" type="SubtotalType" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.66.1 DynamicRows.Grouping

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicRows.Grouping** element specifies the expressions by which to group the data. This element **MUST** be specified and is of type [Grouping](#).

Following is the parent element of the **DynamicRows.Grouping** element.

Parent elements
<a href="#">DynamicRows</a>

The following is the XML Schema definition of the **DynamicRows.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" />
```

## 2.66.2 DynamicRows.ReportItems

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicRows.ReportItems** element specifies a collection of the elements of the row header layout. This element **MUST** be specified and is of type [ReportItems](#). The **ReportItems** collection **MUST** contain exactly one **report item**. The **Top**, **Left**, **Height**, and **Width** elements for this report item are ignored, and their values are interpreted as follows:

- Position: "0, 0"
- Size: "100%, 100%"

Following is the parent element of the **DynamicRows.ReportItems** element.

Parent elements
<a href="#">DynamicRows</a>

The following is the XML Schema definition of the **DynamicRows.ReportItems** element.

```
<xsd:element name="ReportItems" type="ReportItemsType" />
```

## 2.66.3 DynamicRows.Sorting

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicRows.Sorting** element specifies the expressions by which to sort rows. This element is optional and is of type [Sorting](#).

Following is the parent element of the **DynamicRows.Sorting** element.

Parent elements
<a href="#">DynamicRows</a>

The following is the XML Schema definition of the **DynamicRows.Sorting** element.

```
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
```

## 2.66.4 DynamicRows.Subtotal

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicRows.Subtotal** element specifies that an automatic subtotal row is included in the rendering of a [Matrix](#). This element is optional and is of type [Subtotal](#).

Following is the parent element of the **DynamicRows.Subtotal** element.

<b>Parent elements</b>
------------------------

<a href="#">DynamicRows</a>
-----------------------------

The following is the XML Schema definition of the **DynamicRows.Subtotal** element.

```
<xsd:element name="Subtotal" type="SubtotalType" minOccurs="0" />
```

## 2.66.5 DynamicRows.Visibility

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicRows.Visibility** element specifies the presentational presence of the **dynamic rows** for this grouping. This element is optional and is of type [Visibility](#).

Following is the parent element of the **DynamicRows.Visibility** element.

<b>Parent elements</b>
------------------------

<a href="#">DynamicRows</a>
-----------------------------

The following is the XML Schema definition of the **DynamicRows.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

## 2.67 StaticRows

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **StaticRows** element specifies a collection of row header regions with a fixed set of rows in a [Matrix](#). Only one [RowGrouping](#) in each matrix can be static. If the **StaticRows** element is specified, there MUST be at least one and there can be more than one [StaticRow](#) in the **StaticRows** collection.

The following are the parent and child elements of the **StaticRows** element.

<b>Parent elements</b>
------------------------

<a href="#">RowGrouping</a>
-----------------------------

<b>Child elements</b>
-----------------------

<a href="#">StaticRows.StaticRow</a>
--------------------------------------

The following is the XML Schema definition of the **StaticRows** element.

```
<xsd:complexType name="StaticRowsType">
  <xsd:sequence>
    <xsd:element name="StaticRow" type="StaticRowType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

## 2.67.1 StaticRows.StaticRow

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **StaticRows.StaticRow** element specifies a member of the collection of **static row** headers for a row grouping in a [Matrix](#). This element **MUST** be specified and is of type [StaticRow](#). More than one **StaticRows.StaticRow** element can be specified under a single [StaticRows](#) parent element.

Following is the parent element of the **StaticRows.StaticRow** element.

Parent elements
<a href="#">StaticRows</a>

The following is the XML Schema definition of the **StaticRows.StaticRow** element.

```
<xsd:element name="StaticRow" type="StaticRowType" maxOccurs="unbounded" />
```

## 2.68 StaticRow

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **StaticRow** element specifies a fixed row header region in a [Matrix](#).

The following are the parent and child elements of the **StaticRow** element.

Parent elements
<a href="#">StaticRows</a>

Child elements
<a href="#">StaticRow.ReportItems</a>

The following is the XML Schema definition of the **StaticRow** element.

```
<xsd:complexType name="StaticRowType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.68.1 StaticRow.ReportItems

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **StaticRow.ReportItems** element specifies a collection of the **report items** of the row header layout in a [Matrix](#). This element **MUST** be specified and is of type [ReportItems](#). The **ReportItems** collection **MUST** contain exactly one report item. The **Top**, **Left**, **Height**, and **Width** elements for this report item are ignored, and their values are interpreted as follows:

- Position: "0, 0"

- Size: "100%, 100%"

Following is the parent element of the **StaticRow.ReportItems** element.

Parent elements
<a href="#">StaticRow</a>

The following is the XML Schema definition of the **StaticRow.ReportItems** element.

```
<xsd:element name="ReportItems" type="ReportItemsType" />
```

## 2.69 Table

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **Table** element specifies a tabular grouped layout of a **data region**. This element MUST have at least one of the following child elements: [Details](#), [Header](#), or [Footer](#).

The following are the parent elements, attributes, and child elements of the **Table** element.

Parent elements
<a href="#">ReportItems</a>
<a href="#">CustomReportItem.AltReportItem</a>

Attributes
<a href="#">Table.Name</a>

Child elements
<a href="#">Table.Style</a>
<a href="#">Table.Action</a>
<a href="#">Table.LinkToChild</a>
<a href="#">Table.Bookmark</a>
<a href="#">Table.CustomProperties</a>
<a href="#">Table.DataElementName</a>
<a href="#">Table.DataElementOutput</a>
<a href="#">Table.Label</a>
<a href="#">Table.Height</a>
<a href="#">Table.Left</a>
<a href="#">Table.RepeatWith</a>

Child elements
<a href="#">Table.ToolTip</a>
<a href="#">Table.Top</a>
<a href="#">Table.Visibility</a>
<a href="#">Table.Width</a>
<a href="#">Table.ZIndex</a>
<a href="#">Table.KeepTogether</a>
<a href="#">Table.NoRows</a>
<a href="#">Table.PageBreakAtEnd</a>
<a href="#">Table.PageBreakAtStart</a>
<a href="#">Table.DataSetName</a>
<a href="#">Table.Filters</a>
<a href="#">Table.DetailDataCollectionName</a>
<a href="#">Table.DetailDataElementName</a>
<a href="#">Table.DetailDataElementOutput</a>
<a href="#">Table.Details</a>
<a href="#">Table.FillPage</a>
<a href="#">Table.Footer</a>
<a href="#">Table.Header</a>
<a href="#">Table.TableColumns</a>
<a href="#">Table.TableGroups</a>

The following is the XML Schema definition of the **Table** element.

```
<xsd:complexType name="TableType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
    <xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

```

<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:element name="TableColumns" type="TableColumnsType" />
<xsd:element name="Header" type="HeaderType" minOccurs="0" />
<xsd:element name="TableGroups" type="TableGroupsType" minOccurs="0" />
<xsd:element name="Details" type="DetailsType" minOccurs="0" />
<xsd:element name="Footer" type="FooterType" minOccurs="0" />
<xsd:element name="FillPage" type="xsd:boolean" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="DetailDataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DetailDataCollectionName" type="xsd:string" minOccurs="0" />
<xsd:element name="DetailDataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.69.1 Table.Name

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.Name** attribute specifies a unique identifier for a [Table](#). This attribute **MUST** be specified. The value of the **Table.Name** attribute **MUST** be a **NormalizedString** that is a case-sensitive **CLS-compliant identifier** [UTR15]. This value **MUST** be unique among the set of all report item and scope names.

Following is the parent element of the **Table.Name** attribute.

Parent elements
Table

The following is the XML Schema definition of the **Table.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.69.2 Table.Style

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)



The **Table.Style** element specifies the style information for a [Table](#). This element is optional and is of type [Style](#).

Following is the parent element of the **Table.Style** element.

Parent elements
<a href="#">Table</a>

The following is the XML Schema definition of the **Table.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.69.3 Table.Action

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.Action** element is ignored.

Following is the parent element of the **Table.Action** element.

Parent elements
<a href="#">Table</a>

The following is the XML Schema definition of the **Table.Action** element.

```
<xsd:element name="Action" type="ActionType" minOccurs="0" />
```

### 2.69.4 Table.LinkToChild

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.LinkToChild** element is ignored.

Following is the parent element of the **Table.LinkToChild** element.

Parent elements
<a href="#">Table</a>

The following is the XML Schema definition of the **Table.LinkToChild** element.

```
<xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
```

### 2.69.5 Table.Bookmark

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.Bookmark** element specifies a bookmark for a [Table](#) that can be linked to via a bookmark action. This element is optional. If the **Table.Bookmark** element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Table.Bookmark** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.Bookmark** element.

```
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
```

## 2.69.6 Table.CustomProperties

**Applies to** [RDL 2005/01](#)

The **Table.CustomProperties** element specifies extended information in a collection of name/value pairs. This element is optional and is of type [CustomProperties](#).

Following is the parent element of the **Table.CustomProperties** element.

Parent elements
<a href="#">Table</a>

The following is the XML Schema definition of the **Table.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

In [RDL 2003/10](#), the equivalent element of **Table.CustomProperties** is **Table.Custom**, which is of type [Custom](#).

## 2.69.7 Table.DataElementName

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.DataElementName** element specifies the name to use for the **data element** of a **table** in a **data rendering**. This element is optional. If this element is present, its value **MUST** be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as the value of the **Name** attribute of the [Table](#).

Following is the parent element of the **Table.DataElementName** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

## 2.69.8 Table.DataElementOutput

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.DataElementOutput** element specifies whether a [Table](#) is included in a **data rendering**. This element is optional. If this element is present, its value **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). The value of the **Table.DataElementOutput** element **MUST** be one of the following:

**Auto** (default): Specifies the default setting for how the table will be included in a data rendering.

**Output**: Specifies that the table is included in a data rendering.

**NoOutput**: Specifies that the table is not included in a data rendering.

**ContentsOnly**: Specifies that the table is included in a data rendering.

The value of **Table.DataElementOutput** is interpreted as "NoOutput" if the following conditions are true:

- [Table.Visibility](#) is specified for the table.
- [Visibility.Hidden](#) is specified for the **Table.Visibility** element.
- [Visibility.ToggleItem](#) for the **Table.Visibility** element is not specified.
- The value of **Visibility.Hidden** for the **Table.Visibility** element is the **String** literal value "true".

Alternatively, the value of **Table.DataElementOutput** is interpreted as "NoOutput" if the following conditions are true:

- The table is contained within a table column.
- [TableColumn.Visibility](#) is specified for the [TableColumn](#).
- **Visibility.Hidden** is specified for the **TableColumn.Visibility** element.
- **Visibility.ToggleItem** for the **TableColumn.Visibility** element is not specified.
- The value of **Visibility.Hidden** for the **TableColumn.Visibility** element is the **String** literal value "true".

Otherwise, the value of **Table.DataElementOutput** is interpreted as "Output".

If this element is not specified, its value is interpreted as "Auto".

Following is the parent element of the **Table.DataElementOutput** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.69.9 Table.Label

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.Label** element specifies a document map label to identify a [Table](#) within the rendered report. This element is optional.

If the **Table.Label** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **Variant**. If the expression returns NULL, no item is added to the document map. This element is ignored if the table is contained within a page header or page footer.

Following is the parent element of the **Table.Label** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

### 2.69.10 **Table.Height**

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.Height** element specifies the height of a [Table](#). This element is optional. If this element is present, its value MUST be a non-negative [RdlSize](#).

If specified, the value of **Table.Height** after performing the above validations will always be ignored. Whether specified or not, the value of **Table.Height** MUST instead be derived from the sum of the heights of the component parts of the table, which are the header, detail, and footer rows.

Following is the parent element of the **Table.Height** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="0" />
```

### 2.69.11 **Table.Left**

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.Left** element specifies the distance of a [Table](#) from the left of the table's container. This element is optional. If this element is specified, its value MUST be a non-negative [RdlSize](#). If the **Table.Left** element is not specified, the value of this element is interpreted as 0.

Following is the parent element of the **Table.Left** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.Left** element.

```
<xsd:element name="Left" type="SizeType" minOccurs="0" />
```

### 2.69.12 Table.RepeatWith

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.RepeatWith** element MUST NOT be specified.

Following is the parent element of the **Table.RepeatWith** element.

Parent elements
<a href="#">Table</a>

The following is the XML Schema definition of the **Table.RepeatWith** element.

```
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
```

### 2.69.13 Table.ToolTip

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.ToolTip** element specifies a tooltip for a [Table](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **Table.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, the value of the element is interpreted as NULL.

Following is the parent element of the **Table.ToolTip** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

### 2.69.14 Table.Top

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.Top** element specifies the distance of a [Table](#) from the top of the table's container. This element is optional. If this element is present, its value MUST be a non-negative [RdlSize](#) constant. If this element is not present, the value of the element is interpreted as 0.

Following is the parent element of the **Table.Top** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.Top** element.

```
<xsd:element name="Top" type="SizeType" minOccurs="0" />
```

### 2.69.15 Table.Visibility

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.Visibility** element specifies the presentational presence of a [Table](#). This element is optional and is of type [Visibility](#).

Following is the parent element of the **Table.Visibility** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

### 2.69.16 Table.Width

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.Width** element specifies the width of a [Table](#). This element is optional. If this element is present, its value MUST be a non-negative [RdlSize](#).

If specified, the value of **Table.Width**, after the data type is validated, will always be ignored. Whether specified or not, the value of **Table.Width** MUST instead be derived from the sum of the widths of the component parts of the table, which are the columns.

Following is the parent element of the **Table.Width** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="0" />
```

### 2.69.17 Table.ZIndex

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.ZIndex** element specifies the **ZIndex** of a [Table](#). This element is optional. If this element is specified, its value MUST be an **UnsignedInt**. If this element is not specified, its value is interpreted as 0.

Following is the parent element of the **Table.ZIndex** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.69.18 **Table.KeepTogether**

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.KeepTogether** element specifies whether the entire contents of a [Table](#), including all repeated sections, are to be kept together on one [Page](#) if possible. This element is optional. The value of this element MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not specified, its value is interpreted as false.

Following is the parent element of the **Table.KeepTogether** element.

Parent elements
<a href="#">List</a>

The following is the XML Schema definition of the **Table.KeepTogether** element.

```
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
```

### 2.69.19 **Table.NoRows**

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.NoRows** element specifies text to render instead of the table layout when no rows of data are available for a [Table](#). This element is optional. If this element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Table.NoRows** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.NoRows** element.

```
<xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
```

### 2.69.20 **Table.PageBreakAtEnd**

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.PageBreakAtEnd** element specifies that a renderer inserts a page break at the end of this [Table](#). This element is optional. The value of the **Table.PageBreakAtEnd** element MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not specified, its value is interpreted as false.

Following is the parent element of the **Table.PageBreakAtEnd** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.PageBreakAtEnd** element.

```
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
```

## 2.69.21 Table.PageBreakAtStart

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Table.PageBreakAtStart** element specifies that the renderer adds a page break at the start of this [Table](#). This element is optional. The value of **Table.PageBreakAtStart** MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not specified, its value is interpreted as false.

Following is the parent element of the **Table.PageBreakAtStart** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.PageBreakAtStart** element.

```
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
```

## 2.69.22 Table.DataSetName

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Table.DataSetName** element specifies the name of the [DataSet](#) to use to bind data to a [Table](#). This element is optional. If this element is present, its value MUST be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is the value of the **Name** attribute of a **DataSet** element contained within the [Report](#).

This element MUST be specified if both of the following statements are true:

- The table is not contained within another data region.
- There is more than one dataset specified for the report.

This element is ignored for a table that is contained within another data region. A table MUST NOT be specified in a report if there is no **DataSet** element specified in the containing report. If this element is not present, its value is interpreted as the value of the **DataSetName** of the first ancestor data region the table is contained within, or the name of the single **DataSet** specified for the report.

Following is the parent element of the **Table.DataSetName** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
```

## 2.69.23 Table.Filters

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Table.Filters** element specifies a collection of [Filter](#) elements to be applied to the data for each row of the table. This element is optional and is of type [Filters](#).

Following is the parent element of the **Table.Filters** element.



<b>Parent elements</b>
<a href="#">Table</a>

The following is the XML Schema definition of the **Table.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

## 2.69.24 Table.DetailDataCollectionName

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.DetailDataCollectionName** element specifies the name to use for the data element in a **data rendering** that is the collection of all instances of the detail data of an instance of this [Table](#). This element is optional.

If the **Table.DetailDataCollectionName** element is present, its value MUST be a **CLS-compliant identifier** [\[UTR15\]](#).

If this element is not present, its value is interpreted as the string that is the concatenation of the value of [Table.DetailDataElementName](#) and the string "\_Collection". If

**Table.DetailDataElementName** is not specified, its default value is used, and the string "Detail\_Collection" is used as the default value of **Table.DetailDataCollectionName** if **Table.DetailDataCollectionName** is not specified.

If there is a grouping specified for the details, the **Table.DetailDataCollectionName** element is ignored and the value of [Grouping.DataCollectionName](#) is used for the name of the data element instead.

Following is the parent element of the **Table.DetailDataCollectionName** element.

<b>Parent elements</b>
<a href="#">Table</a>

The following is the XML Schema definition of the **Table.DetailDataCollectionName** element.

```
<xsd:element name="DetailDataCollectionName" type="xsd:string" minOccurs="0" />
```

## 2.69.25 Table.DetailDataElementName

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.DetailDataElementName** element specifies the name to use for the **data element** of the detail data of an instance of this [Table](#) in a **data rendering**. This element is optional. If this element is present, its value MUST be a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as "Detail".

If a grouping is specified for the details, the **Table.DetailDataElementName** element is ignored and the value of [Grouping.DataElementName](#) is used as the name of the data element instead.

Following is the parent element of the **Table.DetailDataElementName** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.DetailDataElementName** element.

```
<xsd:element name="DetailDataElementName" type="xsd:string" minOccurs="0" />
```

### 2.69.26 Table.DetailDataElementOutput

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.DetailDataElementOutput** element specifies whether the data in the details is included in a **data rendering** of the [Table](#). This element is optional. If specified, the value of the **Table.DetailDataElementOutput** element MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1). The specified value for this element MUST be one of the following:

**Output** (Default): Specifies that the detail data is included in a data rendering.

**NoOutput**: Specifies that the detail data is not included in a data rendering.

**ContentsOnly**: Specifies that the detail data is included in a data rendering.

If this element is not specified, its value is interpreted as "Output".

If a grouping is specified for the detail data, the **Table.DetailDataElementOutput** element is ignored, and the value of [Grouping.DataElementOutput](#) is used to determine whether the data in the details is included in a data rendering.

Following is the parent element of the **Table.DetailDataElementOutput** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.DetailDataElementOutput** element.

```
<xsd:element name="DetailDataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.69.27 Table.Details

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.Details** element specifies the collection of detail rows for a [Table](#). This element is optional and is of type [Details](#).

Following is the parent element of the **Table.Details** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.Details** element.

```
<xsd:element name="Details" type="DetailsType" minOccurs="0" />
```

### 2.69.28 Table.FillPage

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.FillPage** element is ignored.

Following is the parent element of the **Table.FillPage** element.

Parent elements
<a href="#">Table</a>

The following is the XML Schema definition of the **Table.FillPage** element.

```
<xsd:element name="FillPage" type="xsd:boolean" minOccurs="0" />
```

### 2.69.29 Table.Footer

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.Footer** element specifies the collection of footer rows in a [Table](#). This element is optional and is of type [Footer](#).

Following is the parent element of the **Table.Footer** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.Footer** element.

```
<xsd:element name="Footer" type="FooterType" minOccurs="0" />
```

### 2.69.30 Table.Header

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.Header** element specifies the collection of header rows in a [Table](#). This element is optional and is of type [Header](#).

Following is the parent element of the **Table.Header** element.

Parent elements
Table

The following is the XML Schema definition of the **Table.Header** element.

```
<xsd:element name="Header" type="HeaderType" minOccurs="0" />
```

### 2.69.31 Table.TableColumns

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.TableColumns** element specifies the collection of columns in a [Table](#). This element is of type [TableColumns](#). The **Table.TableColumns** element MUST be specified.

Following is the parent element of the **Table.TableColumns** element.

Parent elements
<a href="#">Table</a>

The following is the XML Schema definition of the **Table.TableColumns** element.

```
<xsd:element name="TableColumns" type="TableColumnsType" />
```

### 2.69.32 Table.TableGroups

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Table.TableGroups** element specifies the collection of **table groups** consisting of **group expressions**, group headers, and group footers for a table. This element is optional and is of type [TableGroups](#).

Following is the parent element of the **Table.TableGroups** element.

Parent elements
<a href="#">Table</a>

The following is the XML Schema definition of the **Table.TableGroups** element.

```
<xsd:element name="TableGroups" type="TableGroupsType" minOccurs="0" />
```

## 2.70 Details

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Details** element specifies the details rows for a [Table](#).

The following are the parent and child elements of the **Details** element.

Parent elements
<a href="#">Table</a>

Child elements
<a href="#">Details.Grouping</a>
<a href="#">Details.Sorting</a>
<a href="#">Details.TableRows</a>
<a href="#">Details.Visibility</a>

The following is the XML Schema definition of the **Details** element.

```
<xsd:complexType name="DetailsType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TableRows" type="TableRowsType" />
    <xsd:element name="Grouping" type="GroupingType" minOccurs="0" />
    <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.70.1 Details.Grouping

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Details.Grouping** element specifies the expressions by which to group the detail data in a [Table](#). This element is optional and is of type [Grouping](#). If the **Details.Grouping** element is not specified, the detail data is not grouped.

Following is the parent element of the **Details.Grouping** element.

Parent elements
<a href="#">Details</a>

The following is the XML Schema definition of the **Details.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" minOccurs="0" />
```

### 2.70.2 Details.Sorting

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Details.Sorting** element specifies the expressions by which to sort the detail data in a [Table](#). This element is optional and is of type [Sorting](#).

Following is the parent element of the **Details.Sorting** element.

Parent elements
<a href="#">Details</a>

The following is the XML Schema definition of the **Details.Sorting** element.

```
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
```

### 2.70.3 Details.TableRows

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Details.TableRows** element specifies the detail rows in a [Table](#). The detail rows MUST NOT contain any data regions in any of their nested [TableCells](#). This element MUST be specified and is of type [TableRows](#).

Following is the parent element of the **Details.TableRows** element.

Parent elements
<a href="#">Details</a>

The following is the XML Schema definition of the **Details.TableRows** element.

```
<xsd:element name="TableRows" type="TableRowsType" />
```

### 2.70.4 Details.Visibility

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Details.Visibility** element specifies the presentational presence of the detail rows in a [Table](#). This element is optional and is of type [Visibility](#).

Following is the parent element of the **Details.Visibility** element.

Parent elements
<a href="#">Details</a>

The following is the XML Schema definition of the **Details.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

## 2.71 TableRows

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableRows** element specifies a collection of **table** rows as an ordered list. If the **TableRows** element is specified, there MUST be at least one and there can be more than one [TableRow](#) in the **TableRows** collection.

The following are the parent and child elements of the **TableRows** element.

Parent elements
<a href="#">Details</a>
<a href="#">Header</a>
<a href="#">Footer</a>

Child elements
----------------

<a href="#">TableRows.TableRow</a>
------------------------------------

The following is the XML Schema definition of the **TableRows** element.

```
<xsd:complexType name="TableRowsType">  
  <xsd:sequence>  
    <xsd:element name="TableRow" type="TableRowType" maxOccurs="unbounded" />  
  </xsd:sequence>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

### 2.71.1 TableRows.TableRow

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableRows.TableRow** element specifies a single row of cells in a [Table](#). This element **MUST** be specified. More than one **TableRows.TableRow** element can be specified under a single [TableRows](#) parent element.

The **TableRows.TableRow** element is of type [TableRow](#).

Following is the parent element of the **TableRows.TableRow** element.

Parent elements
-----------------

<a href="#">TableRows</a>
---------------------------

The following is the XML Schema definition of the **TableRows.TableRow** element.

```
<xsd:element name="TableRow" type="TableRowType" maxOccurs="unbounded" />
```

### 2.72 TableRow

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableRow** element specifies a row of cells in a table data region.

The following are the parent and child elements of the **TableRow** element.

Parent elements
-----------------

<a href="#">TableRows</a>
---------------------------

Child elements
----------------

<a href="#">TableRow.Height</a>
---------------------------------

<a href="#">TableRow.TableCells</a>
-------------------------------------

<a href="#">TableRow.Visibility</a>
-------------------------------------

The following is the XML Schema definition of the **TableRow** element.

```
<xsd:complexType name="TableRowType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TableCells" type="TableCellsType" />
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.72.1 TableRow.Height

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableRow.Height** element specifies the height of a row in a [Table](#). The **TableRow.Height** element MUST be specified. The value of this element MUST be a non-negative [RDLSize](#).

Following is the parent element of the **TableRow.Height** element.

Parent elements
<a href="#">TableRow</a>

The following is the XML Schema definition of the **TableRow.Height** element.

```
<xsd:element name="Height" type="SizeType" />
```

### 2.72.2 TableRow.TableCells

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableRow.TableCells** element specifies the contents of a row in a [Table](#). The **TableRow.TableCells** element MUST be specified. This element is of type [TableCells](#).

Following is the parent element of the **TableRow.TableCells** element.

Parent elements
<a href="#">TableRow</a>

The following is the XML Schema definition of the **TableRow.TableCells** element.

```
<xsd:element name="TableCells" type="TableCellsType" />
```

### 2.72.3 TableRow.Visibility

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableRow.Visibility** element specifies the presentational presence of a row in a [Table](#). This element is optional and is of type [Visibility](#).



Following is the parent element of the **TableRow.Visibility** element.

Parent elements
<a href="#">TableRow</a>

The following is the XML Schema definition of the **TableRow.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

## 2.73 TableCells

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableCells** element specifies a collection of cells in a row of a [Table](#) data region. If the **TableCells** element is specified, there MUST be at least one and there can be more than one [TableCell](#) in the **TableCells** collection. There MUST be one **TableCell** per column in the table except that [TableCell.ColSpan](#) can allow multiple columns to be covered by one cell.

The following are the parent and child elements of the **TableCells** element.

Parent elements
<a href="#">TableRow</a>

Child elements
<a href="#">TableCells.TableCell</a>

The following is the XML Schema definition of the **TableCells** element.

```
<xsd:complexType name="TableCellsType">
  <xsd:sequence>
    <xsd:element name="TableCell" type="TableCellType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.73.1 TableCells.TableCell

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableCells.TableCell** element specifies the contents of a single cell in a [Table](#) data region. This element MUST be specified and is of type [TableCell](#). More than one **TableCells.TableCell** element can be specified under a single [TableCells](#) parent element.

Following is the parent element of the **TableCells.TableCell** element.

Parent elements
TableCells

The following is the XML Schema definition of the **TableCells.TableCell** element.

```
<xsd:element name="TableCell" type="TableCellType" maxOccurs="unbounded" />
```

## 2.74 TableCell

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableCell** element specifies the contents of a cell in a [Table](#) data region.

The following are the parent and child elements of the **TableCell** element.

Parent elements
<a href="#">TableCells</a>

Child elements
<a href="#">TableCell.ColSpan</a>
<a href="#">TableCell.ReportItems</a>

The following is the XML Schema definition of the **TableCell** element.

```
<xsd:complexType name="TableCellType">  
  <xsd:choice minOccurs="1" maxOccurs="unbounded">  
    <xsd:element name="ReportItems" type="ReportItemsType" />  
    <xsd:element name="ColSpan" type="xsd:unsignedInt" minOccurs="0" />  
    <xsd:any namespace="##other" processContents="skip" />  
  </xsd:choice>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

### 2.74.1 TableCell.ColSpan

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableCell.ColSpan** element specifies the number of columns that the cell spans. This element is optional. If the **TableCell.ColSpan** element is present, its value MUST be an **UnsignedInt** whose value MUST be greater than or equal to 1. If this element is not present, its value is interpreted as 1.

The sum of the **TableCell.ColSpan** values that are descendants of each [TableRow](#) MUST equal the number of [TableColumn](#) elements in the [TableColumns](#) collection. This sum of values includes counting the default value of "1" for each [TableCell](#) element that does not specify a **TableCell.ColSpan** element.

Following is the parent element of the **TableCell.ColSpan** element.

Parent elements
<a href="#">TableCell</a>

The following is the XML Schema definition of the **TableCell.ColSpan** element.

```
<xsd:element name="ColSpan" type="xsd:unsignedInt" minOccurs="0" />
```

## 2.74.2 TableCell.ReportItems

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableCell.ReportItems** element specifies an element of the **report layout** as a collection of **report items** (such as **lists**, **text boxes**, and **lines**) that is contained within a cell of a [Table](#).

This element MUST be specified and is of type [ReportItems](#). The **ReportItems** collection MUST contain exactly one report item. The **Top**, **Left**, **Height**, and **Width** elements for this report item are ignored, and their values are interpreted as follows:

- Position: "0, 0"
- Size: "100%, 100%"

Page breaks on report items inside a [TableCell](#) are ignored.

Following is the parent element of the **TableCell.ReportItems** element.

Parent elements
<a href="#">TableCell</a>

The following is the XML Schema definition of the **TableCell.ReportItems** element.

```
<xsd:element name="ReportItems" type="ReportItemsType" />
```

## 2.75 Footer

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Footer** element specifies the footer rows for a [Table](#) or [TableGroup](#).

The following are the parent and child elements of the **Footer** element.

Parent elements
<a href="#">Table</a>
<a href="#">TableGroup</a>

Child elements
<a href="#">Footer.RepeatOnNewPage</a>
<a href="#">Footer.TableRows</a>

The following is the XML Schema definition of the **Footer** element.

```
<xsd:complexType name="FooterType">  
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
```

```

    <xsd:element name="TableRows" type="TableRowsType" />
    <xsd:element name="RepeatOnNewPage" type="xsd:boolean" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.75.1 Footer.RepeatOnNewPage

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Footer.RepeatOnNewPage** element specifies whether rendered footers are repeated on every [Page](#) where the footer's parent [Table](#) or [TableGroup](#) is rendered. This element is optional. If this element is specified, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not specified, its value is interpreted as false.

Following is the parent element of the **Footer.RepeatOnNewPage** element.

Parent elements
<a href="#">Footer</a>

The following is the XML Schema definition of the **Footer.RepeatOnNewPage** element.

```

<xsd:element name="RepeatOnNewPage" type="xsd:boolean" minOccurs="0" />

```

### 2.75.2 Footer.TableRows

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Footer.TableRows** element specifies the footer rows for a [Table](#) or [TableGroup](#). This element MUST be specified and is of type [TableRows](#).

Following is the parent element of the **Footer.TableRows** element.

Parent elements
<a href="#">Footer</a>

The following is the XML Schema definition of the **Footer.TableRows** element.

```

<xsd:element name="TableRows" type="TableRowsType" />

```

## 2.76 Header

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Header** element specifies the header rows for a [Table](#) or [TableGroup](#).

The following are the parent and child elements of the **Header** element.

Parent elements
Table
TableGroup

Child elements
<a href="#">Header.FixedHeader</a>
<a href="#">Header.RepeatOnNewPage</a>
<a href="#">Header.TableRows</a>

The following is the XML Schema definition of the **Header** element.

```
<xsd:complexType name="HeaderType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TableRows" type="TableRowsType" />
    <xsd:element name="FixedHeader" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="RepeatOnNewPage" type="xsd:boolean" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.76.1 Header.FixedHeader

**Applies to** [RDL 2005/01](#)

The **Header.FixedHeader** element specifies that the header remains visible when the [Table](#) is partially scrolled off the [Page](#). This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false. If this element is present, its value MUST NOT be true if this is the header for a [TableGroup](#).

Following is the parent element of the **Header.FixedHeader** element.

Parent elements
<a href="#">Header</a>

The following is the XML Schema definition of the **Header.FixedHeader** element.

```
<xsd:element name="FixedHeader" type="xsd:boolean" minOccurs="0" />
```

### 2.76.2 Header.RepeatOnNewPage

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Header.RepeatOnNewPage** element specifies whether rendered headers are repeated on every [Page](#) where the header's parent [Table](#) or [TableGroup](#) is rendered. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Header.RepeatOnNewPage** element.

Parent elements
<a href="#">Header</a>

The following is the XML Schema definition of the **Header.RepeatOnNewPage** element.

```
<xsd:element name="RepeatOnNewPage" type="xsd:boolean" minOccurs="0" />
```

### 2.76.3 Header.TableRows

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Header.TableRows** element specifies the header rows for a [Table](#) or [TableGroup](#). This element MUST be specified and is of type [TableRows](#).

Following is the parent element of the **Header.TableRows** element.

Parent elements
<a href="#">Header</a>

The following is the XML Schema definition of the **Header.TableRows** element.

```
<xsd:element name="TableRows" type="TableRowsType" />
```

### 2.77 TableColumns

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **TableColumns** element specifies the collection of columns in a Table. If the **TableColumns** element is specified, there MUST be at least one and there can be more than one [TableColumn](#) in the **TableColumns** collection. There MUST be one **TableColumn** per column in the table.

The following are the parent and child elements of the **TableColumns** element.

Parent elements
<a href="#">Table</a>

Child elements
<a href="#">TableColumns.TableColumn</a>

The following is the XML Schema definition of the **TableColumns** element.

```
<xsd:complexType name="TableColumnsType">
  <xsd:sequence>
    <xsd:element name="TableColumn" type="TableColumnType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

## 2.77.1 TableColumns.TableColumn

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableColumns.TableColumn** element specifies a column in a [Table](#). This element MUST be specified and is of type [TableColumn](#). More than one **TableColumns.TableColumn** element can be specified under a single [TableColumns](#) parent element.

Following is the parent element of the **TableColumns.TableColumn** element.

Parent elements
<a href="#">TableColumns</a>

The following is the XML Schema definition of the **TableColumns.TableColumn** element.

```
<xsd:element name="TableColumn" type="TableColumnType" maxOccurs="unbounded" />
```

## 2.78 TableColumn

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableColumn** element specifies a column in a [Table](#).

The following are the parent and child elements of the **TableColumn** element.

Parent elements
<a href="#">TableColumns</a>

Child elements
<a href="#">TableColumn.FixedHeader</a>
<a href="#">TableColumn.Visibility</a>
<a href="#">TableColumn.Width</a>

The following is the XML Schema definition of the **TableColumn** element.

```
<xsd:complexType name="TableColumnType">  
  <xsd:choice minOccurs="1" maxOccurs="unbounded">  
    <xsd:element name="Width" type="SizeType" />  
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />  
    <xsd:element name="FixedHeader" type="xsd:boolean" minOccurs="0" />  
    <xsd:any namespace="##other" processContents="skip" />  
  </xsd:choice>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

### 2.78.1 TableColumn.FixedHeader

**Applies to** [RDL 2005/01](#)

The **TableColumn.FixedHeader** element specifies whether this column remains visible when the [Table](#) is partially scrolled off the [Page](#). The fixed header columns in a table MUST be contiguous and MUST include the first or last column in the table.

This element is optional. The value of the **TableColumn.FixedHeader** element MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not specified, its value is interpreted as false.

Following is the parent element of the **TableColumn.FixedHeader** element.

Parent elements
<a href="#">TableColumn</a>

The following is the XML Schema definition of the **TableColumn.FixedHeader** element.

```
<xsd:element name="FixedHeader" type="xsd:boolean" minOccurs="0" />
```

### 2.78.2 TableColumn.Visibility

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **TableColumn.Visibility** element specifies the presentational presence of a column in a [Table](#). This element is optional. The **TableColumn.Visibility** element is of type [Visibility](#).

Following is the parent element of the **TableColumn.Visibility** element.

Parent elements
<a href="#">TableColumn</a>

The following is the XML Schema definition of the **TableColumn.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

### 2.78.3 TableColumn.Width

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **TableColumn.Width** element specifies the width of a column. This element MUST be specified. The value of this element MUST be a non-negative [RdlSize](#).

Following is the parent element of the **TableColumn.Width** element.

Parent elements
<a href="#">TableColumn</a>

The following is the XML Schema definition of the **TableColumn.Width** element.

```
<xsd:element name="Width" type="SizeType" />
```

## 2.79 TableGroups

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*



The **TableGroups** element specifies the collection of groups in a [Table](#). If the **TableGroups** element is present, there MUST be at least one and there can be more than one [TableGroup](#) in the **TableGroups** collection.

The following are the parent and child elements of the **TableGroups** element.

Parent elements
<a href="#">Table</a>

Child elements
<a href="#">TableGroups.TableGroup</a>

The following is the XML Schema definition of the **TableGroups** element.

```
<xsd:complexType name="TableGroupsType">
  <xsd:sequence>
    <xsd:element name="TableGroup" type="TableGroupType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.79.1 TableGroups.TableGroup

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **TableGroups.TableGroup** element specifies a group consisting of **group expressions**, group headers, and/or group footers in a [Table](#). This element MUST be specified. More than one **TableGroups.TableGroup** element can be specified under a single [TableGroups](#) parent element.

The **TableGroups.TableGroup** element is of type [TableGroup](#).

Following is the parent element of the **TableGroups.TableGroup** element.

Parent elements
<a href="#">TableGroups</a>

The following is the XML Schema definition of the **TableGroups.TableGroup** element.

```
<xsd:element name="TableGroup" type="TableGroupType" maxOccurs="unbounded" />
```

### 2.80 TableGroup

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **TableGroup** element specifies a group in a [Table](#) data region.

The following are the parent and child elements of the **TableGroup** element.

Parent elements
<a href="#">TableGroups</a>

Child elements
<a href="#">TableGroup.Footer</a>
<a href="#">TableGroup.Grouping</a>
<a href="#">TableGroup.Header</a>
<a href="#">TableGroup.Sorting</a>
<a href="#">TableGroup.Visibility</a>

The following is the XML Schema definition of the **TableGroup** element.

```
<xsd:complexType name="TableGroupType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Grouping" type="GroupingType" />
    <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
    <xsd:element name="Header" type="HeaderType" minOccurs="0" />
    <xsd:element name="Footer" type="FooterType" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.80.1 TableGroup.Footer

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableGroup.Footer** element specifies a footer for a [TableGroup](#). This element is optional and is of type [Footer](#).

Following is the parent element of the **TableGroup.Footer** element.

Parent elements
TableGroup

The following is the XML Schema definition of the **TableGroup.Footer** element.

```
<xsd:element name="Footer" type="FooterType" minOccurs="0" />
```

### 2.80.2 TableGroup.Grouping

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableGroup.Grouping** element specifies the expressions by which to group the data in a [TableGroup](#). This element **MUST** be specified and is of type [Grouping](#).

Following is the parent element of the **TableGroup.Grouping** element.

Parent elements
TableGroup

The following is the XML Schema definition of the **TableGroup.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" />
```

### 2.80.3 TableGroup.Header

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableGroup.Header** element specifies a header for a [TableGroup](#). This element is optional and is of type [Header](#).

Following is the parent element of the **TableGroup.Header** element.

Parent elements
TableGroup

The following is the XML Schema definition of the **TableGroup.Header** element.

```
<xsd:element name="Header" type="HeaderType" minOccurs="0" />
```

### 2.80.4 TableGroup.Sorting

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableGroup.Sorting** element specifies the expressions by which to sort the data in a [TableGroup](#). This element is optional and is of type [Sorting](#).

Following is the parent element of the **TableGroup.Sorting** element.

Parent elements
TableGroup

The following is the XML Schema definition of the **TableGroup.Sorting** element.

```
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
```

### 2.80.5 TableGroup.Visibility

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **TableGroup.Visibility** element specifies the presentational presence of a [TableGroup](#). This element is optional and is of type [Visibility](#).

Following is the parent element of the **TableGroup.Visibility** element.

Parent elements
TableGroup

The following is the XML Schema definition of the **TableGroup.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

## 2.81 Group

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Group** element defines expressions by which to categorize data and properties of the resulting data categorization.

If an element of type **Group** does not have exactly one child [GroupExpressions.GroupExpression](#) element, the [Group.Parent](#) element MUST NOT be specified.

The following are the parent elements, attribute, and child elements of the **Group** element.

Parent elements
<a href="#">ChartMember</a>
<a href="#">DataMember</a>
<a href="#">GaugeMember</a>
<a href="#">MapMember</a>
<a href="#">TablixMember</a>

Attributes
<a href="#">Group.Name</a>

Child elements
<a href="#">Group.DataElementName</a>
<a href="#">Group.DataElementOutput</a>
<a href="#">Group.DocumentMapLabel</a>
<a href="#">Group.DomainScope</a>
<a href="#">Group.Filters</a>
<a href="#">Group.GroupExpressions</a>
<a href="#">Group.PageBreak</a>
<a href="#">Group.PageName</a>
Group.Parent
<a href="#">Group.ReGroupExpressions</a>
<a href="#">Group.Variables</a>

**Applies to** [RDL 2011/01](#)

Child elements
<a href="#">Group.DataSetName</a>
<a href="#">Group.NaturalGroup</a>
<a href="#">Group.Relationship</a>

The following is the XML Schema definition of the **Group** element in RDL 2008/01.

```
<xsd:complexType name="GroupType" >
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="GroupExpressions" type="GroupExpressionsType" minOccurs="0" />
    <xsd:element name="ReGroupExpressions" type="GroupExpressionsType" minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="Parent" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Variables" type="VariablesType" minOccurs="0" />
    <xsd:element name="DomainScope" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **Group** element in RDL 2010/01 and RDL 2016/01.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:complexType name="GroupType" >
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="GroupExpressions" type="GroupExpressionsType" minOccurs="0" />
    <xsd:element name="ReGroupExpressions" type="GroupExpressionsType" minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
    <xsd:element name="PageName" type="xsd:string" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="Parent" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Variables" type="VariablesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
</xsd:complexType>
```

```

</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.81.1 Group.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Group.Name** attribute specifies a unique identifier for a [Group](#). This attribute **MUST** be specified and its value **MUST** be a **NormalizedString** that is a case-sensitive **CLS-compliant identifier** [UTR15]. This value **MUST** be unique among all scope names.

Following is the parent element of the **Group.Name** attribute.

Parent elements
Group

The following is the XML Schema definition of the **Group.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.81.2 Group.DataElementName

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Group.DataElementName** element specifies the name to use for the **data element** of a [Group](#) in a **data rendering**. This element is optional and **MUST NOT** be specified more than once.

If the **Group.DataElementName** element is specified, its value **MUST** be a [String](#) that is a **CLS-compliant identifier** [UTR15] that is unique within the parent element of the data element. If the **Group.DataElementName** element is not specified, its value is interpreted as the value of the [Group.Name](#) attribute of the parent **Group** element.

Following is the parent element of the **Group.DataElementName** element.

Parent elements
Group

The following is the XML Schema definition of the **Group.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.81.3 Group.DataElementOutput

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Group.DataElementOutput** element specifies whether a [Group](#) is included in a **data rendering**. This element is optional and **MUST NOT** be specified more than once. If this element is specified, its value **MUST** be one of the following:

**Output** (default): The **Group** data appears in the data rendering output.

**NoOutput:** The **Group** data does not appear in the data rendering output.

If this element is not specified, its value is interpreted as "Output".

Following is the parent element of the **Group.DataElementOutput** element

Parent elements
Group

The following is the XML Schema definition of the **Group.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

#### 2.81.4 Group.DataSetName

**Applies to** [RDL 2011/01](#)

The **Group.DataSetName** element specifies which [DataSet](#) to use for this [Group](#). The **Group.DataSetName** element is optional and MUST NOT be specified more than once.

If the **Group.DataSetName** element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#). If the **Group.DataSetName** element is not present, its value is interpreted as the **DataSetName** of the containing scope (**DataRegion** or **Group**).

Following is the parent element of the **Group.DataSetName** element.

Parent elements
Group

The following is the XML Schema definition of the **Group.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
```

#### 2.81.5 Group.DocumentMapLabel

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Group.DocumentMapLabel** element specifies a **document map** label to identify a [Group](#) within the rendered report. This element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [String](#).

Following is the parent element of the **Group.DocumentMapLabel** element.

Parent elements
Group

The following is the XML Schema definition of the **Group.DocumentMapLabel** element.

```
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
```

### 2.81.6 Group.DomainScope

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Group.DomainScope** element specifies the name of the scope (data region or group) in which to evaluate the **group expressions**. The **Group.DomainScope** element is optional and MUST NOT be specified more than once. If the **Group.DomainScope** element is specified, it is of type [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). The element **Group.DomainScope** is not allowed within a **detail group**.

Following is the parent element of the **Group.DomainScope** element.

Parent elements
<a href="#">Group</a>

The following is the XML Schema definition of the **Group.DomainScope** element.

```
<xsd:element name="DomainScope" type="xsd:string" minOccurs="0" />
```

### 2.81.7 Group.Filters

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Group.Filters** element specifies a collection of [Filter](#) elements for a [Group](#). The **Group.Filters** element is optional and MUST NOT be specified more than once. If the **Group.Filters** element is specified, it is of type [Filters](#).

Following is the parent element of the **Group.Filters** element.

Parent elements
<a href="#">Group</a>

The following is the XML Schema definition of the **Group.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

### 2.81.8 Group.GroupExpressions

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Group.GroupExpressions** element specifies a collection of [GroupExpressions.GroupExpression](#) elements of a [Group](#). The **Group.GroupExpressions** element is optional and MUST NOT be specified more than once. If this element is specified, it MUST be of type [GroupExpressions](#). If the **Group.GroupExpressions** element is not specified, the group is a **detail group**.

Following is the parent element of the **Group.GroupExpressions** element.



Parent elements
Group

The following is the XML Schema definition of the **Group.GroupExpressions** element.

```
<xsd:element name="GroupExpressions" type="GroupExpressionsType" minOccurs="0" />
```

### 2.81.9 Group.NaturalGroup

*Applies to [RDL 2011/01](#)*

The **Group.NaturalGroup** element specifies whether the associated dataset has data grouped by the group expression of this [Group](#). The **Group.NaturalGroup** element is optional. If this element is present, its value MUST be a Boolean ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false. [<13><14><15>](#)

Following is the parent element of the **Group.NaturalGroup** element.

Parent elements
Group

The following is the XML Schema definition of the **Group.NaturalGroup** element.

```
<xsd:element name="NaturalGroup" type="xsd:boolean" minOccurs="0" />
```

### 2.81.10 Group.PageBreak

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Group.PageBreak** element specifies a [PageBreak](#) element of a [Group](#). The **Group.PageBreak** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type **PageBreak**. [<16>](#)

Following is the parent element of the **Group.PageBreak** element.

Parent elements
Group

The following is the XML Schema definition of the **Group.PageBreak** element.

```
<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
```

### 2.81.11 Group.PageName

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **Group.PageName** element specifies value to use for the name of a paginated page. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Group.PageName** element.

Parent elements
<a href="#">Group</a>

The following is the XML Schema definition of the **Group.PageName** element.

```
<xsd:element name="PageName" type="xsd:string" minOccurs="0" />
```

### 2.81.12 Group.Parent

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Group.Parent** element specifies an expression that identifies a parental hierarchy for a [Group](#). The **Group.Parent** element is optional and MUST NOT be specified more than once. If this element is specified, its value MUST be a [String](#) or an expression that evaluates to a **Variant**. If a **Group** element does not have exactly one child [GroupExpressions.GroupExpression](#) element, the **Group.Parent** element MUST NOT be specified.

Following is the parent element of the **Group.Parent** element.

Parent elements
Group

The following is the XML Schema definition of the **Group.Parent** element.

```
<xsd:element name="Parent" type="xsd:string" minOccurs="0" />
```

### 2.81.13 Group.ReGroupExpressions

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Group.ReGroupExpressions** element is ignored.

Following is the parent element of the **Group.ReGroupExpressions** element.

Parent elements
<a href="#">Group</a>

The following is the XML Schema definition of the **Group.ReGroupExpressions** element.

```
<xsd:element name="ReGroupExpressions" type="GroupExpressionsType" minOccurs="0" />
```

### 2.81.14 Group.Relationship

*Applies to [RDL 2011/01](#)*

The **Group.Relationship** element specifies a **relationship** to use for correlating data in this [Group](#) with the data in the containing scope. The **Group.Relationship** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [Relationship](#). This element is ignored if the [DataSet](#) for this **Group** is the same as the **DataSet** for each containing scope.

Following is the parent element of the **Group.Relationship** element.

Parent elements
Group

The following is the XML Schema definition of the **Group.Relationship** element.

```
<xsd:element name="Relationship" type="RelationshipType" minOccurs="0" />
```

### 2.81.15 Group.Variables

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Group.Variables** element specifies a [Variables](#) element of a [Group](#). The **Group.Variables** element is optional and MUST NOT be specified more than once. If the **Group.Variables** element is specified, it is of type **Variables**.

Following is the parent element of the **Group.Variables** element.

Parent elements
Group

The following is the XML Schema definition of the **Group.Variables** element.

```
<xsd:element name="Variables" type="VariablesType" minOccurs="0" />
```

## 2.82 GroupExpressions

The **GroupExpressions** element defines a collection of [GroupExpressions.GroupExpression](#) elements for a grouping.

The following are the parent and child elements of the **GroupExpressions** element.

Parent elements
<a href="#">Group</a>
<a href="#">Grouping</a>

Child elements
GroupExpressions.GroupExpression

The following is the XML Schema definition of the **GroupExpressions** element.

```
<xsd:complexType name="GroupExpressionsType">
  <xsd:sequence>
    <xsd:element name="GroupExpression" type="xsd:string" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.82.1 GroupExpressions.GroupExpression

The **GroupExpressions.GroupExpression** element specifies an expression by which to group data. This element MUST be specified at least once and can be specified more than once. The value of this element MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **VARIANT**.

The value of the **GroupExpressions.GroupExpression** element MUST NOT include any **aggregate functions** other than the [RowNumber](#) aggregate function. If the RowNumber aggregate function is used, it MUST reference the immediately containing scope.

Following is the parent element of the **GroupExpressions.GroupExpression** element.

Parent elements
<a href="#">GroupExpressions</a>

The following is the XML Schema definition of the **GroupExpressions.GroupExpression** element.

```
<xsd:element name="GroupExpression" type="xsd:string" maxOccurs="unbounded" />
```

### 2.83 SortExpressions

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **SortExpressions** element defines a collection of [SortExpression](#) elements.

The following are the parent and child elements of the **SortExpressions** element.

Parent elements
<a href="#">ChartMember</a>
<a href="#">DataMember</a>
<a href="#">GaugeMember</a>
<a href="#">TablixMember</a>
<a href="#">Chart</a>
<a href="#">Tablix</a>
<a href="#">GaugePanel</a>
<a href="#">CustomData</a>

Child elements
<a href="#">SortExpressions.SortExpression</a>

The following is the XML Schema definition of the **SortExpressions** element in RDL 2008/01.

```
<xsd:complexType name="SortExpressionsType">
  <xsd:sequence>
    <xsd:element name="SortExpression" type="SortExpressionType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

```

    <xsd:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **SortExpressions** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="SortExpressionsType">
  <xsd:sequence>
    <xsd:element name="SortExpression" type="SortExpressionType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.83.1 SortExpressions.SortExpression

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **SortExpressions.SortExpression** element specifies an expression with which to order data. This element **MUST** be specified at least once and can be specified more than once, and is of type [SortExpression](#).

Following is the parent element of the **SortExpressions.SortExpression** element.

<b>Parent elements</b>
<a href="#">SortExpressions</a>

The following is the XML Schema definition of the **SortExpressions.SortExpression** element.

```

<xsd:element name="SortExpression" type="SortExpressionType" minOccurs="1"
  maxOccurs="unbounded" />

```

### 2.84 SortExpression

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **SortExpression** element defines an expression used in sorting.

The following are the parent and child elements of the **SortExpression** element.

<b>Parent elements</b>
<a href="#">SortExpressions</a>

<b>Child elements</b>
<a href="#">SortExpression.Direction</a>
<a href="#">SortExpression.Value</a>

## Applies to [RDL 2011/01](#)

<b>Child elements</b>
<a href="#">SortExpression.NaturalSort</a>
<a href="#">SortExpression.DeferredSort</a>

The following is the XML Schema definition of the **SortExpression** element in RDL 2008/01.

```
<xsd:complexType name="SortExpressionType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="Direction" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Ascending" />
          <xsd:enumeration value="Descending" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **SortExpression** element in RDL 2010/01 and RDL 2016/01.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:complexType name="SortExpressionType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="Direction" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Ascending" />
          <xsd:enumeration value="Descending" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.84.1 SortExpression.Direction

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **SortExpression.Direction** element specifies the sort order of the [SortExpression](#) element. The **SortExpression.Direction** element is optional and MUST NOT be specified more than once. If this element is specified, its value MUST be one of the following:

**Ascending** (default): Data is sorted in ascending order.

**Descending**: Data is sorted in descending order.

If the **SortExpression.Direction** element is not specified, its value is interpreted as "Ascending".

Following is the parent element of the **SortExpression.Direction** element.

Parent elements
SortExpression

The following is the XML Schema definition of the **SortExpression.Direction** element.

```
<xsd:element name="Direction" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Ascending" />
      <xsd:enumeration value="Descending" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.84.2 SortExpression.NaturalSort

**Applies to** [RDL 2011/01](#)

The **SortExpression.NaturalSort** element specifies whether associated data set has data sorted by this [SortExpression](#). The **SortExpression.NaturalSort** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false. [<17><18><19><20><21>](#)

Following is the parent element of the **SortExpression.NaturalSort** element.

Parent elements
SortExpression

The following is the XML Schema definition of the **SortExpression.NaturalSort** element.

```
<xsd:element name="NaturalSort" type="xsd:boolean" minOccurs="0" />
```

### 2.84.3 SortExpression.Value

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **SortExpression.Value** element specifies an expression that results in a datum by which to order. This element MUST be specified and MUST NOT be specified more than once. The value of this element MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **VARIANT**.

The value of this element MUST NOT include the [RunningValue](#) or [RowNumber](#) **aggregate functions** and MUST NOT include a reference to any report item.

Following is the parent element of the **SortExpression.Value** element.

Parent elements
<a href="#">SortExpression</a>

The following is the XML Schema definition of the **SortExpression.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1" />
```

## 2.84.4 SortExpression.DeferredSort

**Applies to** [RDL 2011/01](#)

The **SortExpression.DeferredSort** element specifies whether an external component performs deferred sorting. The **SortExpression.DeferredSort** element is optional and MUST NOT be specified more than once. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **SortExpression.DeferredSort** element.

Parent elements
<a href="#">SortExpression</a>

The following is the XML Schema definition of the **SortExpression.DeferredSort** element.

```
<xsd:element name="DeferredSort" type="xsd:boolean" minOccurs="0" />
```

## 2.85 Grouping

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Grouping** element specifies the expressions by which to categorize data in the parent element (see the following **Parent elements** table) and the properties of the resulting data categorization.

If an element of type **Grouping** does not have exactly one child [GroupExpressions.GroupExpression](#) element in the [Grouping.GroupExpressions](#) collection, the [Grouping.Parent](#) element MUST NOT be specified.

The following are the parent elements, attributes, and child elements of the **Grouping** element.

Parent elements
<a href="#">DataGrouping</a>
<a href="#">DynamicColumns</a>
<a href="#">DynamicRows</a>
<a href="#">List</a>
<a href="#">TableGroup</a>
<a href="#">Details</a>
<a href="#">DynamicCategories</a>
<a href="#">DynamicSeries</a>

Attributes
<a href="#">Grouping.Name</a>



Child elements
<a href="#">Grouping.CustomProperties</a>
<a href="#">Grouping.DataCollectionName</a>
<a href="#">Grouping.DataElementName</a>
<a href="#">Grouping.DataElementOutput</a>
<a href="#">Grouping.Filters</a>
Grouping.GroupExpressions
<a href="#">Grouping.Label</a>
<a href="#">Grouping.PageBreakAtEnd</a>
<a href="#">Grouping.PageBreakAtStart</a>
Grouping.Parent

The following is the XML Schema definition of the **Grouping** element.

```
<xsd:complexType name="GroupingType" >
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="GroupExpressions" type="GroupExpressionsType" />
    <xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="Parent" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataCollectionName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip">
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.85.1 Grouping.Name

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Grouping.Name** attribute specifies a unique identifier for a [Grouping](#). This attribute MUST be specified. The value of the **Grouping.Name** attribute MUST be a [NormalizedString](#) that is a case-sensitive **CLS-compliant identifier** [UTR15]. This value MUST be unique among the set of all grouping, [DataSet](#), and data region names.

Following is the parent element of the **Grouping.Name** attribute.

Parent elements
Grouping

The following is the XML Schema definition of the **Grouping.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

## 2.85.2 Grouping.CustomProperties

**Applies to** [RDL 2005/01](#)

The **Grouping.CustomProperties** element specifies extended information in a collection of name/value pairs. This element is optional and is of type [CustomProperties](#).

Following is the parent element of the **Grouping.CustomProperties** element.

Parent elements
<a href="#">Grouping</a>

The following is the XML Schema definition of the **Grouping.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

In [RDL 2003/10](#), the equivalent element of **Grouping.CustomProperties** is **Grouping.Custom**, which is of type [Custom](#).

## 2.85.3 Grouping.DataCollectionName

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Grouping.DataCollectionName** element specifies the name to use for the data element in a **data rendering** that is the collection of all instances of this [Grouping](#). This element is optional.

If this element is present, its value MUST be a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as the string that is the concatenation of the value of [Grouping.DataElementName](#) and the string "\_Collection". If **Grouping.DataElementName** is not specified, its default value is used. In this case, the string that is the concatenation of the value of [Grouping.Name](#) and the string "\_Collection" is used as the default value of **Grouping.DataCollectionName**.

Following is the parent element of the **Grouping.DataCollectionName** element.

Parent elements
Grouping

The following is the XML Schema definition of the **Grouping.DataCollectionName** element.

```
<xsd:element name="DataCollectionName" type="xsd:string" minOccurs="0" />
```

## 2.85.4 Grouping.DataElementName

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Grouping.DataElementName** element specifies the name to use for the **data element** that is an instance of this group in a **data rendering**. This element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as the value of the **Name** attribute of the [Grouping](#) element.

Following is the parent element of the **Grouping.DataElementName** element.

Parent elements
Grouping

The following is the XML Schema definition of the **Grouping.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

## 2.85.5 Grouping.DataElementOutput

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Grouping.DataElementOutput** element specifies whether a [Grouping](#) is included in a **data rendering**. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

The value of the **Grouping.DataElementOutput** element MUST be one of the following:

**Output** (default): Specifies that the group is included in a data rendering.

**NoOutput**: Specifies that the group is not included in a data rendering.

**ContentsOnly**: Specifies that the group is included in a data rendering.

If this element is not present, its value is interpreted as "Output".

Following is the parent element of the **Grouping.DataElementOutput** element.

Parent elements
Grouping

The following is the XML Schema definition of the **Grouping.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.85.6 Grouping.Filters

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Grouping.Filters** element specifies a collection of [Filter](#) elements to be applied to the data for each instance of a [Grouping](#). This element is optional and is of type [Filters](#).

Following is the parent element of the **Grouping.Filters** element.

Parent elements
Grouping

The following is the XML Schema definition of the **Grouping.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

## 2.85.7 Grouping.GroupExpressions

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Grouping.GroupExpressions** element specifies a collection of [GroupExpressions.GroupExpression](#) elements by which to group the data in a [Grouping](#). This element **MUST** be specified and is of type [GroupExpressions](#).

Following is the parent element of the **Grouping.GroupExpressions** element.

Parent elements
Grouping

The following is the XML Schema definition of the **Grouping.GroupExpressions** element.

```
<xsd:element name="GroupExpressions" type="GroupExpressionsType" />
```

## 2.85.8 Grouping.Label

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Grouping.Label** element specifies a document map label with which to identify a [Grouping](#) in a rendered [Report](#). This element is optional.

If the **Grouping.Label** element is present, its value **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If the expression returns **NULL**, no item is added to the document map. This element is ignored if the group is contained within a page header or page footer.

Following is the parent element of the **Grouping.Label** element.

Parent elements
Grouping

The following is the XML Schema definition of the **Grouping.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

## 2.85.9 Grouping.PageBreakAtEnd

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Grouping.PageBreakAtEnd** element specifies that a renderer inserts a page break at the end of a [Grouping](#). This element is optional.

If the **Grouping.PageBreakAtEnd** element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false. The **Grouping.PageBreakAtEnd** element is not valid for column groupings in [Matrix](#) regions and MUST NOT be specified in this case.

Following is the parent element of the **Grouping.PageBreakAtEnd** element.

Parent elements
<a href="#">Grouping</a>

The following is the XML Schema definition of the **Grouping.PageBreakAtEnd** element.

```
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
```

## 2.85.10 Grouping.PageBreakAtStart

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Grouping.PageBreakAtStart** element specifies that a renderer inserts a page break at the start of a group. This element is optional.

If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false. This element is not valid for column groupings in [Matrix](#) regions and MUST NOT be specified in this case.

Following is the parent element of the **Grouping.PageBreakAtStart** element.

Parent elements
<a href="#">Grouping</a>

The following is the XML Schema definition of the **Grouping.PageBreakAtStart** element.

```
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
```

## 2.85.11 Grouping.Parent

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Grouping.Parent** element specifies an expression that identifies the parent group in a recursive hierarchy. This element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **Variant**. If the [Grouping](#) does not have exactly one child [GroupExpressions.GroupExpression](#) element, this element MUST NOT be specified.

Following is the parent element of the **Grouping.Parent** element.

Parent elements
<a href="#">Grouping</a>

The following is the XML Schema definition of the **Grouping.Parent** element.

```
<xsd:element name="Parent" type="xsd:string" minOccurs="0" />
```

## 2.86 Sorting

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **Sorting** element specifies a collection of [SortBy](#) elements by which to sort groups. If the **Sorting** element is specified, there MUST be at least one and there can be more than one **SortBy** element in the **Sorting** collection. The **Sorting** collection is an ordered list, and the sorting expressions MUST be applied in the specified order.

The following are the parent and child elements of the **Sorting** element.

Parent elements
<a href="#">DataGrouping</a>
<a href="#">DynamicColumns</a>
<a href="#">DynamicRows</a>
<a href="#">List</a>
<a href="#">TableGroup</a>
<a href="#">Details</a>
<a href="#">DynamicCategories</a>
<a href="#">DynamicSeries</a>

Child elements
<a href="#">Sorting.SortBy</a>

The following is the XML Schema definition of the **Sorting** element.

```
<xsd:complexType name="SortingType">  
  <xsd:sequence>  
    <xsd:element name="SortBy" type="SortByType" maxOccurs="unbounded" />  
  </xsd:sequence>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

## 2.86.1 Sorting.SortBy

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **Sorting.SortBy** element specifies an expression by which to sort the data. This element MUST be specified and is of type [SortBy](#). More than one **Sorting.SortBy** element can be specified under a single [Sorting](#) parent element. The **Sorting** collection is an ordered list, and the sorting expressions MUST be applied in the specified order.

Following is the parent element of the **Sorting.SortBy** element.

Parent elements
<a href="#">Sorting</a>

The following is the XML Schema definition of the **Sorting.SortBy** element.

```
<xsd:element name="SortBy" type="SortByType" maxOccurs="unbounded" />
```

## 2.87 SortBy

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **SortBy** element specifies an expression by which to sort groups.

The following are the parent and child elements of the **SortBy** element.

Parent elements
<a href="#">Sorting</a>

Child elements
<a href="#">SortBy.Direction</a>
<a href="#">SortBy.SortExpression</a>

The following is the XML Schema definition of the **SortBy** element.

```
<xsd:complexType name="SortByType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="SortExpression" type="xsd:string" />
    <xsd:element name="Direction" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Ascending" />
          <xsd:enumeration value="Descending" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.87.1 SortBy.Direction

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **SortBy.Direction** element specifies the sort order of the [SortBy](#) element. This element is optional and is of type [String](#) ([XMLSCHEMA2/2] section 3.2.1).

If this element is specified, its value MUST be one of the following:

**Ascending** (default): Specifies that the sort is in ascending order.

**Descending**: Specifies that the sort is in descending order.

If this element is not present, its value is interpreted as "Ascending".

Following is the parent element of the **SortBy.Direction** element.

Parent elements
<a href="#">SortBy</a>

The following is the XML Schema definition of the **SortBy.Direction** element.

```
<xsd:element name="Direction" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Ascending" />
      <xsd:enumeration value="Descending" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.87.2 SortBy.SortExpression

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **SortBy.SortExpression** element specifies an expression by which to sort the data. This element MUST be specified, and its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **VARIANT**.

Following is the parent element of the **SortBy.SortExpression** element.

Parent elements
<a href="#">SortBy</a>

The following is the XML Schema definition of the **SortBy.SortExpression** element.

```
<xsd:element name="SortExpression" type="xsd:string" />
```

## 2.88 Chart

The **Chart** element specifies a collection of [ChartArea](#) elements to be drawn as a single **data visualization data region**. In [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#), the **Chart** element MUST contain exactly one [Chart.ChartCategoryHierarchy](#) element and exactly one [Chart.ChartSeriesHierarchy](#) element.



The following are the parent elements of the **Chart** element.

Parent elements
<a href="#">ReportItems</a>
<a href="#">CustomReportItem.AltReportItem</a>

The following are additional parent elements of the **Chart** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Parent elements
<a href="#">CellContents</a>

The following are the attributes and child elements of the **Chart** element.

Attributes
<a href="#">Chart.Name</a>

Child elements
<a href="#">Chart.Bookmark</a>
<a href="#">Chart.ChartData</a>
<a href="#">Chart.CustomProperties</a>
<a href="#">Chart.DataElementName</a>
<a href="#">Chart.DataElementOutput</a>
<a href="#">Chart.DataSetName</a>
<a href="#">Chart.Filters</a>
<a href="#">Chart.Height</a>
<a href="#">Chart.Left</a>
<a href="#">Chart.Palette</a>
<a href="#">Chart.Style</a>
<a href="#">Chart.ToolTip</a>
<a href="#">Chart.Top</a>
<a href="#">Chart.Visibility</a>
<a href="#">Chart.Width</a>
<a href="#">Chart.ZIndex</a>

The following are additional child elements of the **Chart** element in [RDL 2003/10](#) and [RDL 2005/01](#).

Child elements
<a href="#">Chart.Action</a>
<a href="#">Chart.CategoryAxis</a>
<a href="#">Chart.CategoryGroupings</a>
<a href="#">Chart.ChartElementOutput</a>
<a href="#">Chart.KeepTogether</a>
<a href="#">Chart.Label</a>
<a href="#">Chart.Legend</a>
<a href="#">Chart.LinkToChild</a>
<a href="#">Chart.PageBreakAtEnd</a>
<a href="#">Chart.PageBreakAtStart</a>
<a href="#">Chart.PlotArea</a>
<a href="#">Chart.PointWidth</a>
<a href="#">Chart.NoRows</a>
<a href="#">Chart.SeriesGroupings</a>
<a href="#">Chart.Subtype</a>
<a href="#">Chart.ThreeDProperties</a>
<a href="#">Chart.Title</a>
<a href="#">Chart.Type</a>
<a href="#">Chart.ValueAxis</a>

The following are additional child elements of the **Chart** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Child elements
<a href="#">Chart.ActionInfo</a>
<a href="#">Chart.DocumentMapLabel</a>
<a href="#">Chart.RepeatWith</a>
<a href="#">Chart.NoRowsMessage</a>
<a href="#">Chart.PageBreak</a>
<a href="#">Chart.SortExpressions</a>
<a href="#">Chart.ChartAreas</a>
<a href="#">Chart.ChartBorderSkin</a>
Chart.ChartCategoryHierarchy
<a href="#">Chart.ChartCustomPaletteColors</a>

Child elements
<a href="#">Chart.ChartLegends</a>
<a href="#">Chart.ChartNoDataMessage</a>
Chart.ChartSeriesHierarchy
<a href="#">Chart.ChartTitles</a>
<a href="#">Chart.DynamicHeight</a>
<a href="#">Chart.DynamicWidth</a>
<a href="#">Chart.PaletteHatchBehavior</a>

The following are additional child elements of the **Chart** element in RDL 2008/01.

Child elements
<a href="#">Chart.ChartAnnotations</a>
<a href="#">Chart.ChartCodeParameters</a>
<a href="#">Chart.Code</a>
<a href="#">Chart.CodeLanguage</a>

The following are additional child elements of the **Chart** element in RDL 2010/01, and RDL 2016/01.

Child elements
<a href="#">Chart.PageName</a>

**Applies to [RDL 2011/01](#)**

Child elements
<a href="#">Chart.Relationship</a>

The following is the XML Schema definition of the **Chart** element in RDL 2003/10 and RDL 2005/01.

```
<xsd:complexType name="ChartType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Type" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Column" />
          <xsd:enumeration value="Bar" />
          <xsd:enumeration value="Line" />
          <xsd:enumeration value="Pie" />
          <xsd:enumeration value="Scatter" />
          <xsd:enumeration value="Bubble" />
          <xsd:enumeration value="Area" />
          <xsd:enumeration value="Doughnut" />
          <xsd:enumeration value="Stock" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Subtype" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
```

```

        <xsd:enumeration value="Stacked" />
        <xsd:enumeration value="PercentStacked" />
        <xsd:enumeration value="Plain" />
        <xsd:enumeration value="Smooth" />
        <xsd:enumeration value="Exploded" />
        <xsd:enumeration value="Line" />
        <xsd:enumeration value="SmoothLine" />
        <xsd:enumeration value="HighLowClose" />
        <xsd:enumeration value="OpenHighLowClose" />
        <xsd:enumeration value="Candlestick" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="Action" type="ActionType" minOccurs="0" />
<xsd:element name="Top" type="SizeType" minOccurs="0" />
<xsd:element name="Left" type="SizeType" minOccurs="0" />
<xsd:element name="Height" type="SizeType" minOccurs="0" />
<xsd:element name="Width" type="SizeType" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
<xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType"
    minOccurs="0" />
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
<xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:element name="SeriesGroupings" type="SeriesGroupingsType" minOccurs="0" />
<xsd:element name="CategoryGroupings" type="CategoryGroupingsType"
    minOccurs="0" />
<xsd:element name="ChartData" type="ChartDataTypes" minOccurs="0" />
<xsd:element name="Legend" type="LegendType" minOccurs="0" />
<xsd:element name="CategoryAxis" type="CategoryAxisType" minOccurs="0" />
<xsd:element name="ValueAxis" type="ValueAxisType" minOccurs="0" />
<xsd:element name="Title" type="TitleType" minOccurs="0" />
<xsd:element name="PointWidth" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Palette" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Default" />
            <xsd:enumeration value="EarthTones" />
            <xsd:enumeration value="Excel" />
            <xsd:enumeration value="GrayScale" />
            <xsd:enumeration value="Light" />
            <xsd:enumeration value="Pastel" />
            <xsd:enumeration value="SemiTransparent" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="ThreeDProperties" type="ThreeDPropertiesType"
    minOccurs="0" />
<xsd:element name="PlotArea" type="PlotAreaType" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Output" />
            <xsd:enumeration value="NoOutput" />
            <xsd:enumeration value="ContentsOnly" />
            <xsd:enumeration value="Auto" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>

```

```

<xsd:element name="ChartElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **Chart** element in RDL 2008/01.

```

<xsd:complexType name="ChartType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="ChartSeriesHierarchy" type="ChartHierarchyType" />
    <xsd:element name="ChartCategoryHierarchy" type="ChartHierarchyType" />
    <xsd:element name="ChartData" type="ChartDataTypes" minOccurs="0" />
    <xsd:element name="ChartAreas" type="ChartAreasType" minOccurs="0" />
    <xsd:element name="ChartLegends" type="ChartLegendsType" minOccurs="0" />
    <xsd:element name="ChartTitles" type="ChartTitlesType" minOccurs="0" />
    <xsd:element name="DynamicHeight" type="xsd:string" minOccurs="0" />
    <xsd:element name="DynamicWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="Palette" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartCustomPaletteColors"
      type="ChartCustomPaletteColorsType" minOccurs="0" />
    <xsd:element name="PaletteHatchBehavior" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="ChartBorderSkin" type="ChartBorderSkinType" minOccurs="0" />
    <xsd:element name="Code" type="xsd:string" minOccurs="0" />
    <xsd:element name="CodeLanguage" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="CSharp" />
          <xsd:enumeration value="VB" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>

```

```

    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="ChartCodeParameters" type="ChartCodeParametersType"
    minOccurs="0" />
  <xsd:element name="ChartAnnotations" type="ChartAnnotationsType" minOccurs="0" />
  <xsd:element name="ChartNoDataMessage" type="ChartTitleType" minOccurs="0" />
  <xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **Chart** element in RDL 2010/01 and RDL 2016/01.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, 5.6, and 5.7, respectively. For more information about macro- and micro-versioned schemas, see section 2.1.

```

<xsd:complexType name="ChartType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType"
      minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
    <xsd:element name="PageName" type="xsd:string" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="ChartSeriesHierarchy" type="ChartHierarchyType" />
    <xsd:element name="ChartCategoryHierarchy" type="ChartHierarchyType" />
    <xsd:element name="ChartData" type="ChartDataTypes" minOccurs="0" />
    <xsd:element name="ChartAreas" type="ChartAreasType" minOccurs="0" />
    <xsd:element name="ChartLegends" type="ChartLegendsType" minOccurs="0" />
    <xsd:element name="ChartTitles" type="ChartTitlesType" minOccurs="0" />
    <xsd:element name="DynamicHeight" type="xsd:string" minOccurs="0" />
    <xsd:element name="DynamicWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="Palette" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartCustomPaletteColors"
      type="ChartCustomPaletteColorsType" minOccurs="0" />
    <xsd:element name="PaletteHatchBehavior" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="ChartBorderSkin" type="ChartBorderSkinType"

```

```

        minOccurs="0" />
        <xsd:element name="ChartNoDataMessage" type="ChartTitleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.88.1 Chart.Name

The **Chart.Name** attribute specifies the name of a [Chart](#). This attribute **MUST** be specified. The value of the **Chart.Name** attribute **MUST** be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique among data regions, groups, and scope names in the [Report](#).

Following is the parent element of the **Chart.Name** attribute.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.88.2 Chart.Action

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.Action** element is ignored.

Following is the parent element of the **Chart.Action** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.Action** element.

```
<xsd:element name="Action" type="ActionType" minOccurs="0" />
```

### 2.88.3 Chart.ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.ActionInfo** element is ignored.

Following is the parent element of the **Chart.ActionInfo** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

#### 2.88.4 Chart.Bookmark

The **Chart.Bookmark** element specifies a bookmark that can be linked to via a bookmark action. This element is optional. If the **Chart.Bookmark** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **Chart.Bookmark** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.Bookmark** element.

```
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
```

#### 2.88.5 Chart.CategoryAxis

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Chart.CategoryAxis** element specifies the category axis for a [Chart](#). This element is optional. This element is of type [CategoryAxis](#).

Following is the parent element of the **Chart.CategoryAxis** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.CategoryAxis** element.

```
<xsd:element name="CategoryAxis" type="CategoryAxisType" minOccurs="0" />
```

#### 2.88.6 Chart.CategoryGroupings

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Chart.CategoryGroupings** element specifies a set of category (X) groupings for a [Chart](#). This element is optional and is of type [CategoryGroupings](#).

Following is the parent element of the **Chart.CategoryGroupings** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.CategoryGroupings** element.

```
<xsd:element name="CategoryGroupings" type="CategoryGroupingsType"
minOccurs="0" />
```



## 2.88.7 Chart.ChartAnnotations

*Applies to [RDL 2008/01](#)*

The **Chart.ChartAnnotations** element is ignored.

Following is the parent element of the **Chart.ChartAnnotations** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.ChartAnnotations** element.

```
<xsd:element name="ChartAnnotations" type="ChartAnnotationsType" minOccurs="0" />
```

## 2.88.8 Chart.ChartAreas

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Chart.ChartAreas** element specifies the collection of [ChartAreas](#) elements in a [Chart](#). This element is optional. The **Chart.ChartAreas** element is of type **ChartAreas**.

Following is the parent element of the **Chart.ChartAreas** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.ChartAreas** element.

```
<xsd:element name="ChartAreas" type="ChartAreasType" minOccurs="0" />
```

## 2.88.9 Chart.ChartBorderSkin

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Chart.ChartBorderSkin** element specifies the **border skin** of a [Chart](#). The **Chart.ChartBorderSkin** element is optional. This element is of type [ChartBorderSkin](#).

Following is the parent element of the **Chart.ChartBorderSkin** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.ChartBorderSkin** element.

```
<xsd:element name="ChartBorderSkin" type="ChartBorderSkinType" minOccurs="0" />
```

## 2.88.10 Chart.ChartCodeParameters

*Applies to [RDL 2008/01](#)*

The **Chart.ChartCodeParameters** element is ignored.

If this element is specified, errors might occur.

Following is the parent element of the **Chart.ChartCodeParameters** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.ChartCodeParameters** element.

```
<xsd:element name="ChartCodeParameters" type="ChartCodeParametersType" minOccurs="0" />
```

### 2.88.11 Chart.ChartCategoryHierarchy

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.ChartCategoryHierarchy** element specifies the hierarchy of category members in a [Chart](#). This element **MUST** be specified. This element is of type [ChartHierarchy](#).

Following is the parent element of the **Chart.ChartCategoryHierarchy** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.ChartCategoryHierarchy** element.

```
<xsd:element name="ChartCategoryHierarchy" type="ChartHierarchyType" />
```

### 2.88.12 Chart.ChartCustomPaletteColors

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.ChartCustomPaletteColors** element specifies a collection of colors to use for a custom palette. This element is optional.

If this element is present, the value of [Chart.Palette](#) is "Custom". If the **Chart.ChartCustomPaletteColors** element is not present, custom palette colors will not be used. This element is of type [ChartCustomPaletteColors](#).

Following is the parent element of the **Chart.ChartCustomPaletteColors** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.ChartCustomPaletteColors** element.

```
<xsd:element name="ChartCustomPaletteColors" type="ChartCustomPaletteColorsType" minOccurs="0" />
```

### 2.88.13 Chart.ChartData

In [RDL 2003/10](#) and [RDL 2005/01](#), the **Chart.ChartData** element specifies the data values in a [ChartArea](#). In [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#), the **Chart.ChartData** element describes the structure of the data in a [Chart](#). The **Chart.ChartData** element is optional. This element is of type [ChartData](#).

Following is the parent element of the **Chart.ChartData** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.ChartData** element.

```
<xsd:element name="ChartData" type="ChartData" minOccurs="0" />
```

### 2.88.14 Chart.ChartElementOutput

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.ChartElementOutput** element MUST NOT be used.

Following is the parent element of the **Chart.ChartElementOutput** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.ChartElementOutput** element.

```
<xsd:element name="ChartElementOutput" minOccurs="0">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">  
      <xsd:enumeration value="Output" />  
      <xsd:enumeration value="NoOutput" />  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:element>
```

### 2.88.15 Chart.ChartLegends

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.ChartLegends** element specifies the set of chart legends in a [Chart](#). This element is optional. This element is of type [ChartLegends](#).

Following is the parent element of the **Chart.ChartLegends** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.ChartLegends** element.

```
<xsd:element name="ChartLegends" type="ChartLegendsType" minOccurs="0" />
```

### 2.88.16 Chart.ChartNoDataMessage

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.ChartNoDataMessage** element specifies the message to display if a [Chart](#) contains no data. This element is optional. This element is of type [ChartTitle](#).

Following is the parent element of the **Chart.ChartNoDataMessage** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.ChartNoDataMessage** element.

```
<xsd:element name="ChartNoDataMessage" type="ChartTitleType" minOccurs="0" />
```

### 2.88.17 Chart.ChartSeriesHierarchy

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.ChartSeriesHierarchy** element specifies the hierarchy of series members in a [Chart](#). This element **MUST** be specified. This element is of type [ChartHierarchy](#).

Following is the parent element of the **Chart.ChartSeriesHierarchy** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.ChartSeriesHierarchy** element.

```
<xsd:element name="ChartSeriesHierarchy" type="ChartHierarchyType" />
```

### 2.88.18 Chart.ChartTitles

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.ChartTitles** element specifies the collection of chart titles in a [Chart](#). This element is optional. This element is of type [ChartTitles](#).

Following is the parent element of the **Chart.ChartTitles** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.ChartTitles** element.

```
<xsd:element name="ChartTitles" type="ChartTitlesType" minOccurs="0" />
```

### 2.88.19 Chart.Code

Applies to [RDL 2008/01](#)

The **Chart.Code** element is ignored.

Following is the parent element of the **Chart.Code** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.Code** element.

```
<xsd:element name="Code" type="xsd:string" minOccurs="0" />
```

### 2.88.20 Chart.CodeLanguage

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.CodeLanguage** element is ignored.

Following is the parent element of the **Chart.CodeLanguage** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.CodeLanguage** element.

```
<xsd:element name="CodeLanguage" minOccurs="0">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">  
      <xsd:enumeration value="CSharp" />  
      <xsd:enumeration value="VB" />  
    </xsd:restriction>  
  </xsd:simpleType>  
</xsd:element>
```

### 2.88.21 Chart.CustomProperties

The **Chart.CustomProperties** element specifies a set of custom information for a [Chart](#) that is handed to the report rendering component. This element is optional. The **Chart.CustomProperties** element is of type [CustomProperties](#).

Following is the parent element of the **Chart.CustomProperties** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

In section [RDL 2003/10](#), the equivalent element of **Chart.CustomProperties** is **Chart.Custom**, which is of type [Custom](#).

### 2.88.22 Chart.DataElementName

The **Chart.DataElementName** element specifies the name to use for the **data element** or attribute for a [Chart](#). This element is optional. If the **Chart.DataElementName** element is present, its value MUST be a **CLS-compliant identifier** [\[UTR15\]](#).

Following is the parent element of the **Chart.DataElementName** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.88.23 Chart.DataElementOutput

The **Chart.DataElementOutput** element specifies whether an item appears in a **data rendering**. This element is optional. If the **Chart.DataElementOutput** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**Output**: Specifies that the item appears in the data rendering output.

**NoOutput**: Specifies that the item does not appear in the data rendering output.

**ContentsOnly**: Specifies that the item itself does not appear in the data rendering output, but that the contents of the item appear in the data rendering output.

**Auto** (default): Specifies that the item does not appear in the data rendering output if, in [RDL 2003/10](#) and [RDL 2005/01](#), the [Chart.Visibility](#) element has its Hidden property set to true or if, in [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#), the value of the grandchild [Visibility.Hidden](#) element of the [Chart](#) element is present and is set to true. Otherwise, the value of the **Chart.DataElementOutput** element is interpreted as "Output".

If the **Chart.DataElementOutput** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **Chart.DataElementOutput** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

</xsd:element>

## 2.88.24 Chart.DataSetName

The **Chart.DataSetName** element specifies the name of the [DataSet](#) to use for a data region. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

The **Chart.DataSetName** element MUST be specified if both of the following statements are true:

- The [Chart](#) is not contained within another data region.
- More than one dataset is specified for the [Report](#).

If no dataset is specified for the containing report, the chart does not render.

In [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#), if the **Chart** has an ancestor, the value of the **Chart.DataSetName** element is interpreted as the **DataSet.Name** for the containing scope (**DataRegion**, [Group](#), or **Cell**).<22>

Following is the parent element of the **Chart.DataSetName** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
```

## 2.88.25 Chart.DocumentMapLabel

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.DocumentMapLabel** element specifies a **document map** label to identify a [Chart](#) within the client UI to provide a user-friendly label for searching. The **Chart.DocumentMapLabel** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Chart.DocumentMapLabel** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.DocumentMapLabel** element.

```
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
```

## 2.88.26 Chart.DynamicHeight

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.DynamicHeight** element specifies the height to which a [Chart](#) will grow or shrink. The **Chart.DynamicHeight** element is optional. If this element is present, its value MUST be an [RdlSize](#)

or an expression that evaluates to an **RdlSize**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **Chart.DynamicHeight** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.DynamicHeight** element.

```
<xsd:element name="DynamicHeight" type="xsd:string" minOccurs="0" />
```

### 2.88.27 Chart.DynamicWidth

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.DynamicWidth** element specifies the width to which a [Chart](#) will grow or shrink. The **Chart.DynamicWidth** element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **Chart.DynamicWidth** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.DynamicWidth** element.

```
<xsd:element name="DynamicWidth" type="xsd:string" minOccurs="0" />
```

### 2.88.28 Chart.Filters

In [RDL 2003/10](#) and [RDL 2005/01](#), the **Chart.Filters** element specifies a collection of [Filter](#) instances for a [Chart](#). In [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#), the **Chart.Filters** element specifies a set of filters for a Chart. The **Chart.Filters** element is optional. This element is of type [Filters](#).

Following is the parent element of the **Chart.Filters** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

### 2.88.29 Chart.Height

The **Chart.Height** element specifies the height of a [Chart](#). The **Chart.Height** element is optional. If this element is present, its value MUST be a non-negative [RdlSize](#). If the **Chart.Height** element is not present, its value is interpreted as the height of the **chart**'s container, such as [Rectangle](#) or [Body](#), minus the value of the [Chart.Top](#) element, if specified.



Following is the parent element of the **Chart.Height** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="0" />
```

### 2.88.30 Chart.KeepTogether

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.KeepTogether** element MUST NOT be used.

Following is the parent element of the **Chart.KeepTogether** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.KeepTogether** element.

```
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
```

### 2.88.31 Chart.Label

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.Label** element specifies a document map label to identify an instance of a [Chart](#) within the client UI. (This is done to provide a user-friendly label for searching.) This element is optional.

If the **Chart.Label** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Chart.Label** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

### 2.88.32 Chart.Left

The **Chart.Left** element specifies the distance of a [Chart](#) from the left of the **chart**'s container, such as a [Rectangle](#) or [Body](#). The **Chart.Left** element is optional. If this element is present, its value MUST be a non-negative [RdSize](#). If this element is not present, its value is interpreted as 0.

Following is the parent element of the **Chart.Left** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.Left** element.

```
<xsd:element name="Left" type="SizeType" minOccurs="0" />
```

### 2.88.33 Chart.Legend

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.Legend** element specifies the legend for a [Chart](#). This element is optional and is of type [Legend](#).

Following is the parent element of the **Chart.Legend** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.Legend** element.

```
<xsd:element name="Legend" type="LegendType" minOccurs="0" />
```

### 2.88.34 Chart.LinkToChild

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.LinkToChild** element MUST NOT be used.

Following is the parent element of the **Chart.LinkToChild** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.LinkToChild** element.

```
<xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
```

### 2.88.35 Chart.NoRows

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.NoRows** element specifies the title to display if a [Chart](#) contains no data. This element is optional. If this element is present, its value MUST be a [String](#) ([IXMLSCHEMA2/2](#) section 3.2.1).

Following is the parent element of the **Chart.NoRows** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.NoRows** element.

```
<xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
```

### 2.88.36 Chart.NoRowsMessage

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.NoRowsMessage** element is ignored. The [Chart.ChartNoDataMessage](#) element MUST be used instead.

Following is the parent element of the **Chart.NoRowsMessage** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.NoRowsMessage** element.

```
<xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
```

### 2.88.37 Chart.PageBreak

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.PageBreak** element specifies the page break behavior for a [Chart](#). This element is optional. The **Chart.PageBreak** element is of type [PageBreak](#).

Following is the parent element of the **Chart.PageBreak** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.PageBreak** element.

```
<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
```

### 2.88.38 Chart.PageBreakAtEnd

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.PageBreakAtEnd** element specifies whether a page break is inserted after a [Chart](#). This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Chart.PageBreakAtEnd** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.PageBreakAtEnd** element.

```
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
```

### 2.88.39 Chart.PageBreakAtStart

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.PageBreakAtStart** element specifies whether a page break is inserted before a [Chart](#). This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Chart.PageBreakAtStart** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.PageBreakAtStart** element.

```
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
```

### 2.88.40 Chart.PageName

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Chart.PageName** element specifies the value to use for the name of a paginated page. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Chart.PageName** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.PageName** element.

```
<xsd:element name="PageName" type="xsd:string" minOccurs="0" />
```

### 2.88.41 Chart.Palette

The **Chart.Palette** element specifies the **color palette** for the **chart items** in a [Chart](#). This element is optional. If the **Chart.Palette** element is present in [RDL 2003/10](#) or [RDL 2005/01](#), its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). If the **Chart.Palette** element is present in [RDL 2008/01](#), [RDL 2010/01](#), or [RDL 2016/01](#), its value MUST be a **String** or an expression that evaluates to a **String**.

The value of this element MUST be one of the following:

**Custom:** Specifies that the Custom palette is used (not available in RDL 2003/10 and RDL 2005/01).

**Default:** Specifies that the Default palette is used.

**Berry:** Specifies that the Berry palette is used (not available in RDL 2003/10 and RDL 2005/01).

**BrightPastel:** Specifies that the BrightPastel palette is used (not available in RDL 2003/10 and RDL 2005/01).

**Chocolate:** Specifies that the Chocolate palette is used (not available in RDL 2003/10 and RDL 2005/01).

**EarthTones:** Specifies that the EarthTones palette is used.

**Excel:** Specifies that the Excel palette is used.

**Fire:** Specifies that the Fire palette is used (not available in RDL 2003/10 and RDL 2005/01).

**GrayScale:** Specifies that the GrayScale palette is used.

**Light:** Specifies that the Light palette is used.

**Pacific:** Specifies that the Pacific palette is used (not available in RDL 2003/10 and RDL 2005/01).

**PacificLight:** Specifies that the PacificLight palette is used (not available in RDL 2003/10 and RDL 2005/01).

**PacificSemiTransparent:** Specifies that the PacificSemiTransparent palette is used (not available in RDL 2003/10 and RDL 2005/01).

**Pastel:** Specifies that the Pastel palette is used.

**SeaGreen:** Specifies that the SeaGreen palette is used (not available in RDL 2003/10 and RDL 2005/01).

**SemiTransparent:** Specifies that the SemiTransparent palette is used.

If the **Chart.Palette** element is not present, its value is interpreted as "Default".

In RDL 2008/01, RDL 2010/01, and RDL 2016/01, if "Custom" is specified, the chart items are painted white unless the [ChartCustomPaletteColors](#) element is also specified.

Following is the parent element of the **Chart.Palette** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.Palette** element in RDL 2003/10 and RDL 2005/01.

```
<xsd:element name="Palette" minOccurs="0" >
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Default" />
      <xsd:enumeration value="EarthTones" />
      <xsd:enumeration value="Excel" />
      <xsd:enumeration value="GrayScale" />
      <xsd:enumeration value="Light" />
      <xsd:enumeration value="Pastel" />
      <xsd:enumeration value="SemiTransparent" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

The following is the XML Schema definition of the **Chart.Palette** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

```
<xsd:element name="Palette" type="xsd:string" minOccurs="0" />
```

## 2.88.42 Chart.PaletteHatchBehavior

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.PaletteHatchBehavior** element specifies whether hatching is automatically applied to **data points** in a [Chart](#). This element is optional. If the **Chart.PaletteHatchBehavior** element is present, its value **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element **MUST** be one of the following:

**None:** Specifies that no hatching is added to data points.

**Default:** Treated as "None".

**Always:** Specifies that automatic hatching will be applied to all data points (unless [Style.BackgroundHatchType](#) is specified in [ChartDataPoint.Style](#)).

If the **Chart.PaletteHatchBehavior** element is not present, its value is interpreted as "Default".

Following is the parent element of the **Chart.PaletteHatchBehavior** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.PaletteHatchBehavior** element.

```
<xsd:element name="PaletteHatchBehavior" type="xsd:string" minOccurs="0" />
```

## 2.88.43 Chart.PlotArea

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.PlotArea** element specifies properties for a [PlotArea](#). This element is optional. This element is of type **PlotArea**.

Following is the parent element of the **Chart.PlotArea** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.PlotArea** element.

```
<xsd:element name="PlotArea" type="PlotAreaType" minOccurs="0" />
```

## 2.88.44 Chart.PointWidth

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.PointWidth** element specifies the width of bars and columns as a percentage of the available room for each individual column in a [Chart](#). This element is optional.

If the **Chart.PointWidth** element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) and MUST NOT be less than zero. If the value is greater than 100, columns will overlap each other. If the value of this element is 0, the default percentage is used. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **Chart.PointWidth** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.PointWidth** element.

```
<xsd:element name="PointWidth" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.88.45 Chart.Relationship

*Applies to [RDL 2011/01](#)*

The **Chart.Relationship** element specifies a **relationship** to use for correlating data in a [Chart](#) with the data in the containing scope. The **Chart.Relationship** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [Relationship](#). The **Chart.Relationship** element MUST NOT be specified if there is no containing scope.

This element is ignored if the data set for the **Chart** is the same as the dataset for each containing scope.

Following is the parent element of the **Chart.Relationship** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.Relationship** element.

```
<xsd:element name="Relationship" type="RelationshipType" minOccurs="0" />
```

### 2.88.46 Chart.RepeatWith

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Chart.RepeatWith** element MUST NOT be specified.

Following is the parent element of the **Chart.RepeatWith** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.RepeatWith** element.

```
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
```

## 2.88.47 Chart.SeriesGroupings

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.SeriesGroupings** element specifies a set of series groupings for a [Chart](#). This element is optional and is of type [SeriesGrouping](#).

Following is the parent element of the **Chart.SeriesGroupings** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.SeriesGroupings** element.

```
<xsd:element name="SeriesGroupings" type="SeriesGroupingsType" minOccurs="0" />
```

## 2.88.48 Chart.SortExpressions

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Chart.SortExpressions** element is ignored.

Following is the parent element of the **Chart.SortExpressions** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.SortExpressions** element.

```
<xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
```

## 2.88.49 Chart.Style

The **Chart.Style** element specifies the style information for a [Chart](#). This element is optional. The **Chart.Style** element is of type [Style](#).

Following is the parent element of the **Chart.Style** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.88.50 Chart.Subtype

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.Subtype** element specifies the subtype of a [Chart](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:



**Stacked:** The value of the [Chart.Type](#) element can only be "Area", "Bar", or "Column".

**PercentStacked:** The value of the **Chart.Type** element can only be "Area", "Bar", or "Column".

**Plain:** The value of the **Chart.Type** element cannot be "Stock".

**Smooth:** The value of the **Chart.Type** element can only be "Line".

**Exploded:** The value of the **Chart.Type** element can only be "Doughnut" or "Pie".

**Line:** The value of the **Chart.Type** element can only be "Scatter".

**SmoothLine:** The value of the **Chart.Type** element can only be "Scatter".

**HighLowClose:** The value of the **Chart.Type** element can only be "Stock".

**OpenHighLowClose:** The value of the **Chart.Type** element can only be "Stock".

**Candlestick:** The value of the **Chart.Type** element can only be "Stock".

If the value of the **Chart.Type** element is "Stock", the **Chart.Subtype** element MUST be specified. If this element is not present, its value is interpreted as "Plain".

Following is the parent element of the **Chart.Subtype** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.Subtype** element.

```
<xsd:element name="Subtype" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Stacked" />
      <xsd:enumeration value="PercentStacked" />
      <xsd:enumeration value="Plain" />
      <xsd:enumeration value="Smooth" />
      <xsd:enumeration value="Exploded" />
      <xsd:enumeration value="Line" />
      <xsd:enumeration value="SmoothLine" />
      <xsd:enumeration value="HighLowClose" />
      <xsd:enumeration value="OpenHighLowClose" />
      <xsd:enumeration value="Candlestick" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.88.51 Chart.ThreeDProperties

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.ThreeDProperties** element specifies the properties for a 3D chart layout. This element is optional. This element is of type [ThreeDProperties](#).

Following is the parent element of the **Chart.ThreeDProperties** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.ThreeDProperties** element.

```
<xsd:element name="ThreeDProperties" type="ThreeDPropertiesType" minOccurs="0" />
```

### 2.88.52 Chart.Title

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.Title** element specifies a title for a [Chart](#). This element is optional. This element is of type [Title](#).

Following is the parent element of the **Chart.Title** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.Title** element.

```
<xsd:element name="Title" type="TitleType" minOccurs="0" />
```

### 2.88.53 Chart.ToolTip

The **Chart.ToolTip** element specifies the textual label for a [Chart](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **Chart.ToolTip** element is optional. If the **Chart.ToolTip** element is present, its value **MUST** be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. If the **Chart.ToolTip** element is not present, its value is interpreted as an empty string.

Following is the parent element of the **Chart.ToolTip** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.ToolTip** element in [RDL 2003/10](#) and [RDL 2005/01](#).

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

The following is the XML Schema definition of the **Chart.ToolTip** element in [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#).

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

### 2.88.54 Chart.Top

The **Chart.Top** element specifies the distance of a [Chart](#) from the top of the chart's container, such as a [Rectangle](#) or [Body](#). This element is optional. If the **Chart.Top** element is present, its value **MUST** be a non-negative [RdlSize](#). If the **Chart.Top** element is not present, its value is interpreted as 0.

Following is the parent element of the **Chart.Top** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.Top** element.

```
<xsd:element name="Top" type="SizeType" minOccurs="0" />
```

### 2.88.55 Chart.Type

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.Type** element specifies the type of a [Chart](#). This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is one of the following:

**Column:** The value of the [Chart.Subtype](#) element can only be "Plain", "Stacked", or "PercentStacked".

**Bar:** The value of the **Chart.Subtype** element can only be "Plain", "Stacked", or "PercentStacked".

**Line:** The value of the **Chart.Subtype** element can only be "Plain" or "Smooth".

**Pie:** The value of the **Chart.Subtype** element can only be "Plain" or "Exploded".

**Scatter:** The value of the **Chart.Subtype** element can only be "Plain", "Line", or "SmoothLine".

**Bubble:** The value of the **Chart.Subtype** element can only be "Plain".

**Area:** The value of the **Chart.Subtype** element can only be "Plain", "Stacked" or "PercentStacked".

**Doughnut:** The value of the **Chart.Subtype** element can only be "Plain" or "Exploded".

**Stock:** The value of the **Chart.Subtype** element can only be "HighLowClose", "OpenHighLowClose", or "Candlestick".

If the **Chart.Type** element is not present, its value is interpreted as "Column".

Following is the parent element of the **Chart.Type** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.Type** element.

```
<xsd:element name="Type" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Column" />
      <xsd:enumeration value="Bar" />
      <xsd:enumeration value="Line" />
      <xsd:enumeration value="Pie" />
      <xsd:enumeration value="Scatter" />
      <xsd:enumeration value="Bubble" />
      <xsd:enumeration value="Area" />
      <xsd:enumeration value="Doughnut" />
      <xsd:enumeration value="Stock" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.88.56 Chart.ValueAxis

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **Chart.ValueAxis** element specifies the value axis for a [Chart](#). This element is optional. This element is of type [ValueAxis](#).

Following is the parent element of the **Chart.ValueAxis** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.ValueAxis** element.

```
<xsd:element name="ValueAxis" type="ValueAxisType" minOccurs="0" />
```

## 2.88.57 Chart.Visibility

The **Chart.Visibility** element specifies the visibility of a [Chart](#). The **Chart.Visibility** element is optional. This element is of type [Visibility](#).

Following is the parent element of the **Chart.Visibility** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

## 2.88.58 Chart.Width

The **Chart.Width** element specifies the width of a [Chart](#). The **Chart.Width** element is optional. If this element is present, its value MUST be a non-negative [RdlSize](#). If the **Chart.Width** element is not present, its value is interpreted as the width of the chart's container, such as [Rectangle](#) or [Body](#), minus the value of the [Chart.Left](#) element, if that element is specified.

Following is the parent element of the **Chart.Width** element.

Parent elements
Chart

The following is the XML Schema definition of the **Chart.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="0" />
```

## 2.88.59 Chart.ZIndex

The **Chart.ZIndex** element specifies the drawing order of a [Chart](#) within a containing object. Items that have lower indices are drawn first and appear behind items that have higher indices. Items that have equal indices can have an unspecified rendering order.

The **Chart.ZIndex** element is optional. If this element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17). If this element is not present, its value is interpreted as 0. The value of this element MUST be greater than or equal to 0 and less than or equal to 2147483647.

Following is the parent element of the **Chart.ZIndex** element.

Parent elements
<a href="#">Chart</a>

The following is the XML Schema definition of the **Chart.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
```

## 2.89 CategoryAxis

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **CategoryAxis** element specifies a category axis for a [Chart](#).

The following are the parent and child elements of the **CategoryAxis** element.

Parent elements
<a href="#">Chart</a>

Child elements
<a href="#">CategoryAxis.Axis</a>

The following is the XML Schema definition of the **CategoryAxis** element.

```
<xsd:complexType name="CategoryAxisType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Axis" type="AxisType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.89.1 CategoryAxis.Axis

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **CategoryAxis.Axis** element specifies an axis for a [Chart](#). This element is optional and is of type [Axis](#).

Following is the parent element of the **CategoryAxis.Axis** element.

Parent elements
<a href="#">CategoryAxis</a>

The following is the XML Schema definition of the **CategoryAxis.Axis** element.

```
<xsd:element name="Axis" type="AxisType" minOccurs="0" />
```

## 2.90 Axis

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis** element specifies properties for labels, titles, and gridlines along an axis of a [Chart](#).

The following are the parent elements and child elements of the **Axis** element.

Parent elements
<a href="#">CategoryAxis</a>
<a href="#">ValueAxis</a>

Child elements
<a href="#">Axis.CrossAt</a>
<a href="#">Axis.Interlaced</a>
<a href="#">Axis.LogScale</a>
<a href="#">Axis.MajorGridLines</a>
<a href="#">Axis.MajorInterval</a>
<a href="#">Axis.MajorTickMarks</a>
<a href="#">Axis.Margin</a>
<a href="#">Axis.Max</a>
<a href="#">Axis.Min</a>
<a href="#">Axis.MinorGridLines</a>
<a href="#">Axis.MinorInterval</a>
<a href="#">Axis.MinorTickMarks</a>
<a href="#">Axis.Reverse</a>
<a href="#">Axis.Scalar</a>
<a href="#">Axis.Style</a>
<a href="#">Axis.Title</a>
<a href="#">Axis.Visible</a>

The following is the XML Schema definition of the **Axis** element.

```
<xsd:complexType name="AxisType">  
  <xsd:choice minOccurs="0" maxOccurs="unbounded">  
    <xsd:element name="Visible" type="xsd:boolean" minOccurs="0" />  
    <xsd:element name="Style" type="StyleType" minOccurs="0" />  
    <xsd:element name="Title" type="TitleType" minOccurs="0" />  
  </xsd:choice>  
</xsd:complexType>
```

```

<xsd:element name="Margin" type="xsd:boolean" minOccurs="0" />
<xsd:element name="MajorTickMarks" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None" />
      <xsd:enumeration value="Inside" />
      <xsd:enumeration value="Outside" />
      <xsd:enumeration value="Cross" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="MinorTickMarks" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None" />
      <xsd:enumeration value="Inside" />
      <xsd:enumeration value="Outside" />
      <xsd:enumeration value="Cross" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="MajorGridLines" type="MajorGridLinesType" minOccurs="0" />
<xsd:element name="MinorGridLines" type="MinorGridLinesType" minOccurs="0" />
<xsd:element name="MajorInterval" type="xsd:string" minOccurs="0" />
<xsd:element name="MinorInterval" type="xsd:string" minOccurs="0" />
<xsd:element name="Reverse" type="xsd:boolean" minOccurs="0" />
<xsd:element name="CrossAt" type="xsd:string" minOccurs="0" />
<xsd:element name="Interlaced" type="xsd:boolean" minOccurs="0" />
<xsd:element name="Scalar" type="xsd:boolean" minOccurs="0" />
<xsd:element name="Min" type="xsd:string" minOccurs="0" />
<xsd:element name="Max" type="xsd:string" minOccurs="0" />
<xsd:element name="LogScale" type="xsd:boolean" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.90.1 Axis.CrossAt

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.CrossAt** element specifies the value at which an [Axis](#) crosses the other axis. This element is optional. If the **Axis.CrossAt** element is present, its value **MUST** be a **Numeric** or [DateTime](#) ([\[XMLSCHEMA2\]](#) section 3.2.7) constant or an expression that evaluates to a **Variant** of type **Numeric** or **DateTime**. If this element is not present, the behavior is the default behavior of the chart type.

Following is the parent element of the **Axis.CrossAt** element.

Parent elements
Axis

The following is the XML Schema definition of the **Axis.CrossAt** element.

```

<xsd:element name="CrossAt" type="xsd:string" minOccurs="0" />

```

### 2.90.2 Axis.Interlaced

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.Interlaced** element specifies whether strip lines MUST be drawn at an interval of every other grid line for an [Axis](#). If grid lines are not used for the axis, the axis tick marks or labels are used to determine the interval of the interlaced strip lines.

The **Axis.Interlaced** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Axis.Interlaced** element.

Parent elements
Axis

The following is the XML Schema definition of the **Axis.Interlaced** element.

```
<xsd:element name="Interlaced" type="xsd:boolean" minOccurs="0" />
```

### 2.90.3 Axis.LogScale

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Axis.LogScale** element specifies whether an [Axis](#) is logarithmic. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Axis.LogScale** element.

Parent elements
Axis

The following is the XML Schema definition of the **Axis.LogScale** element.

```
<xsd:element name="LogScale" type="xsd:boolean" minOccurs="0" />
```

### 2.90.4 Axis.MajorGridLines

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Axis.MajorGridLines** element specifies the major gridlines for an [Axis](#). This element is optional and is of type [MajorGridLines](#).

Following is the parent element of the **Axis.MajorGridLines** element.

Parent elements
Axis

The following is the XML Schema definition of the **Axis.MajorGridLines** element.

```
<xsd:element name="MajorGridLines" type="MajorGridLinesType" minOccurs="0" />
```



## 2.90.5 Axis.MajorInterval

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.MajorInterval** element specifies the interval between major gridlines and tick marks on an [Axis](#). This element is optional. If the **Axis.MajorInterval** element is present, its value MUST be a **Numeric** or [DateTime](#) ([XMLSCHEMA2](#) section 3.2.7) constant or an expression that evaluates to a **Variant** of type **Numeric** or **DateTime**. If this element is not specified, the axis will determine the interval.

Following is the parent element of the **Axis.MajorInterval** element.

Parent elements
Axis

The following is the XML Schema definition of the **Axis.MajorInterval** element.

```
<xsd:element name="MajorInterval" type="xsd:string" minOccurs="0" />
```

## 2.90.6 Axis.MajorTickMarks

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.MajorTickMarks** element specifies the type of a major tick mark on an [Axis](#). This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) that is one of the following:

**None:** Specifies that tick marks do not appear.

**Inside:** Specifies that tick marks appear inside the axis.

**Outside:** Specifies that tick marks appear outside the axis.

**Cross:** Specifies that tick marks appear across the axis.

If the **Axis.MajorTickMarks** element is not present, its value is interpreted as "None".

Following is the parent element of the **Axis.MajorTickMarks** element.

Parent elements
Axis

The following is the XML Schema definition of the **Axis.MajorTickMarks** element.

```
<xsd:element name="MajorTickMarks" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None" />
      <xsd:enumeration value="Inside" />
      <xsd:enumeration value="Outside" />
      <xsd:enumeration value="Cross" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.90.7 Axis.Margin

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.Margin** element specifies whether an axis margin is created. This element is optional.

The size of the margin MUST be automatically generated based on the axis scale and on the number of data points. If this element is present, its value MUST be a **Boolean** ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, the value of this element is interpreted as false.

Following is the parent element of the **Axis.Margin** element.

Parent elements
<a href="#">Axis</a>

The following is the XML Schema definition of the **Axis.Margin** element.

```
<xsd:element name="Margin" type="xsd:boolean" minOccurs="0" />
```

## 2.90.8 Axis.Max

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.Max** element specifies the maximum value for an [Axis](#). This element is optional. If the **Axis.Max** element is present, its value MUST be a **Numeric** or [DateTime](#) ([\[XMLSCHEMA2\]](#) section 3.2.7) constant or an expression that evaluates to a **Variant** of type **Numeric** or **DateTime**. If this element is not present, the axis calculates its scale automatically.

Following is the parent element of the **Axis.Max** element.

Parent elements
<a href="#">Axis</a>

The following is the XML Schema definition of the **Axis.Max** element.

```
<xsd:element name="Max" type="xsd:string" minOccurs="0" />
```

## 2.90.9 Axis.Min

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.Min** element specifies the minimum value for an [Axis](#). This element is optional. If this element is present, its value MUST be a **Numeric** or [DateTime](#) ([\[XMLSCHEMA2\]](#) section 3.2.7) constant or an expression that evaluates to a **Variant** of type **Numeric** or **DateTime**. If this element is not present, the axis calculates its scale automatically.

Following is the parent element of the **Axis.Min** element.

Parent elements
<a href="#">Axis</a>

The following is the XML Schema definition of the **Axis.Min** element.

```
<xsd:element name="Min" type="xsd:string" minOccurs="0" />
```

### 2.90.10 Axis.MinorGridLines

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.MinorGridLines** element specifies the minor gridlines for an [Axis](#). This element is optional and is of type [MinorGridLines](#).

Following is the parent element of the **Axis.MinorGridLines** element.

Parent elements
Axis

The following is the XML Schema definition of the **Axis.MinorGridLines** element.

```
<xsd:element name="MinorGridLines" type="MinorGridLinesType" minOccurs="0" />
```

### 2.90.11 Axis.MinorInterval

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.MinorInterval** element specifies the interval between minor gridlines and tick marks for an [Axis](#). This element is optional. If this element is present, its value MUST be a **Numeric** or [DateTime](#) ([\[XMLSCHEMA2\]](#) section 3.2.7) constant or an expression that evaluates to a **Variant** of type **Numeric** or **DateTime**. If this element is not present, the axis determines the interval.

Following is the parent element of the **Axis.MinorInterval** element.

Parent elements
Axis

The following is the XML Schema definition of the **Axis.MinorInterval** element.

```
<xsd:element name="MinorInterval" type="xsd:string" minOccurs="0" />
```

### 2.90.12 Axis.MinorTickMarks

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.MinorTickMarks** element specifies the type of a minor tick mark on an [Axis](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**None:** Specifies that tick marks do not appear.

**Inside:** Specifies that tick marks appear inside the axis.

**Outside:** Specifies that tick marks appear outside the axis.

**Cross:** Specifies that tick marks appear across the axis.

If the **Axis.MinorTickMarks** element is not present, its value is interpreted as "None".

Following is the parent element of the **Axis.MinorTickMarks** element.

Parent elements
Axis

The following is the XML Schema definition of the **Axis.MinorTickMarks** element.

```
<xsd:element name="MinorTickMarks" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None" />
      <xsd:enumeration value="Inside" />
      <xsd:enumeration value="Outside" />
      <xsd:enumeration value="Cross" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.90.13 Axis.Reverse

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.Reverse** element specifies whether an [Axis](#) is plotted in the reverse direction. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, the value of this element is interpreted as false.

Following is the parent element of the **Axis.Reverse** element.

Parent elements
Axis

The following is the XML Schema definition of the **Axis.Reverse** element.

```
<xsd:element name="Reverse" type="xsd:boolean" minOccurs="0" />
```

### 2.90.14 Axis.Scalar

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.Scalar** element specifies whether the values along an [Axis](#) are scalar values (that is, numeric or date values) that are displayed on a [Chart](#) in a continuous axis.

The value of this element MUST be false if the axis is a [CategoryAxis](#) and if one of the following conditions is true:

- The [CategoryGrouping](#) element of the chart contains more categories.
- The **CategoryGrouping** element of the chart contains a **static category**.
- The **CategoryGrouping** element of the chart contains a group with multiple grouping expressions.

The **Axis.Scalar** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Axis.Scalar** element.

Parent elements
Axis

The following is the XML Schema definition of the **Axis.Scalar** element.

```
<xsd:element name="Scalar" type="xsd:boolean" minOccurs="0" />
```

### 2.90.15 Axis.Style

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.Style** element specifies the text style properties for axis labels and the line style properties for the axis line. This element is optional and is of type [Style](#).

Following is the parent element of the **Axis.Style** element.

Parent elements
<a href="#">Axis</a>

The following is the XML Schema definition of the **Axis.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.90.16 Axis.Title

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.Title** element specifies a title for an [Axis](#). This element is optional and is of type [Title](#).

Following is the parent element of the **Axis.Title** element.

Parent elements
Axis

The following is the XML Schema definition of the **Axis.Title** element.

```
<xsd:element name="Title" type="TitleType" minOccurs="0" />
```

### 2.90.17 Axis.Visible

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Axis.Visible** element specifies whether axis labels are displayed. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Axis.Visible** element.

Parent elements
<a href="#">Axis</a>

The following is the XML Schema definition of the **Axis.Visible** element.

```
<xsd:element name="Visible" type="xsd:boolean" minOccurs="0" />
```

## 2.91 MajorGridLines

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **MajorGridLines** element specifies style properties for major gridlines along an [Axis](#).

The following are the parent and child elements of the **MajorGridLines** element.

Parent elements
<a href="#">Axis</a>

Child elements
<a href="#">MajorGridLines.ShowGridLines</a>
<a href="#">MajorGridLines.Style</a>

The following is the XML Schema definition of the **MajorGridLines** element.

```
<xsd:complexType name="MajorGridLinesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ShowGridLines" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.91.1 MajorGridLines.ShowGridLines

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **MajorGridLines.ShowGridLines** element specifies whether the gridlines along an [Axis](#) are displayed. This element is optional. If the **MajorGridLines.ShowGridLines** element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, the value of this element is interpreted as false.

The following is the parent element of the **MajorGridLines.ShowGridLines** element.

Parent elements
<a href="#">MajorGridLines</a>

The following is the XML Schema definition of the **MajorGridLines.ShowGridLines** element.

```
<xsd:element name="ShowGridLines" type="xsd:boolean" minOccurs="0" />
```

## 2.91.2 MajorGridLines.Style

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **MajorGridLines.Style** element specifies the line style properties for the gridlines and tick marks along an [Axis](#). This element is optional and is of type [Style](#).

Following is the parent element of the **MajorGridLines.Style** element.

Parent elements
<a href="#">MajorGridLines</a>

The following is the XML Schema definition of the **MajorGridLines.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.92 MinorGridLines

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **MinorGridLines** element specifies style properties for minor gridlines along an [Axis](#).

The following are the parent and child elements of the **MinorGridLines** element.

Parent elements
<a href="#">Axis</a>

Child elements
<a href="#">MinorGridLines.ShowGridLines</a>
<a href="#">MinorGridLines.Style</a>

The following is the XML Schema definition of the **MinorGridLines** element.

```
<xsd:complexType name="MinorGridLinesType">  
  <xsd:choice minOccurs="0" maxOccurs="unbounded">  
    <xsd:element name="ShowGridLines" type="xsd:boolean" minOccurs="0" />  
    <xsd:element name="Style" type="StyleType" minOccurs="0" />  
    <xsd:any namespace="##other" processContents="skip" />  
  </xsd:choice>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

### 2.92.1 MinorGridLines.ShowGridLines

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **MinorGridLines.ShowGridLines** element specifies whether the gridlines for an [Axis](#) are displayed. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([[XMLSCHEMA2/2](#)] section 3.2.2). If this element is not present, the value of this element is interpreted as false.

Following is the parent element of the **MinorGridLines.ShowGridLines** element.

Parent elements
<a href="#">MinorGridLines</a>

The following is the XML Schema definition of the **MinorGridLines.ShowGridLines** element.

```
<xsd:element name="ShowGridLines" type="xsd:boolean" minOccurs="0" />
```

## 2.92.2 MinorGridLines.Style

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **MinorGridLines.Style** element specifies the line style properties for the gridlines and tick marks along an [Axis](#). This element is optional and is of type [Style](#).

Following is the parent element of the **MinorGridLines.Style** element.

Parent elements
<a href="#">MinorGridLines</a>

The following is the XML Schema definition of the **MinorGridLines.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.93 CategoryGroupings

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **CategoryGroupings** element specifies the set [CategoryGrouping](#) instances for a [Chart](#).

The following are the parent and child elements of the **CategoryGroupings** element.

Parent elements
Chart

Child elements
<a href="#">CategoryGroupings.CategoryGrouping</a>

The following is the XML Schema definition of the **CategoryGroupings** element.

```
<xsd:complexType name="CategoryGroupingsType">  
  <xsd:sequence>  
    <xsd:element name="CategoryGrouping" type="CategoryGroupingType" />  
  </xsd:sequence>  
</xsd:complexType>
```



```

        maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.93.1 CategoryGroupings.CategoryGrouping

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **CategoryGroupings.CategoryGrouping** element specifies a collection of category groupings. This element **MUST** be specified and is of type [CategoryGrouping](#).

Following is the parent element of the **CategoryGroupings.CategoryGrouping** element.

Parent elements
<a href="#">CategoryGroupings</a>

The following is the XML Schema definition of the **CategoryGroupings.CategoryGrouping** element.

```

<xsd:element name="CategoryGrouping" type="CategoryGroupingType"
    maxOccurs="unbounded" />

```

## 2.94 CategoryGrouping

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **CategoryGrouping** element specifies a category level for a [Chart](#). The labels for the categories are displayed along the [CategoryAxis](#). The **CategoryGrouping** element **MUST** have either the [DynamicCategories](#) or [StaticCategories](#) element specified, but not both.

The following are the parent and child elements of the **CategoryGrouping** element.

Parent elements
<a href="#">CategoryGroupings</a>

Child elements
<a href="#">CategoryGrouping.DynamicCategories</a>
<a href="#">CategoryGrouping.StaticCategories</a>

The following is the XML Schema definition of the **CategoryGrouping** element.

```

<xsd:complexType name="CategoryGroupingType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="DynamicCategories" type="DynamicCategoriesType"
            minOccurs="0" />
        <xsd:element name="StaticCategories" type="StaticCategoriesType"
            minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />

```

</xsd:complexType>

### 2.94.1 CategoryGrouping.DynamicCategories

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **CategoryGrouping.DynamicCategories** element specifies a collection of **dynamic category** headings for this grouping. This element is optional and is of type [DynamicCategories](#).

Following is the parent element of the **CategoryGrouping.DynamicCategories** element.

Parent elements
<a href="#">CategoryGrouping</a>

The following is the XML Schema definition of the **CategoryGrouping.DynamicCategories** element.

```
<xsd:element name="DynamicCategories" type="DynamicCategoriesType"
  minOccurs="0" />
```

### 2.94.2 CategoryGrouping.StaticCategories

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **CategoryGrouping.StaticCategories** element specifies a collection of **static category** headings for a grouping. This element is optional and is of type [StaticCategories](#).

Following is the parent element of the **CategoryGrouping.StaticCategories** element.

Parent elements
<a href="#">CategoryGrouping</a>

The following is the XML Schema definition of the **CategoryGrouping.StaticCategories** element.

```
<xsd:element name="StaticCategories" type="StaticCategoriesType" minOccurs="0" />
```

## 2.95 DynamicCategories

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **DynamicCategories** element specifies a category header region that repeats with each category group in a rowset of a [Chart](#).

The following are the parent and child elements of the **DynamicCategories** element.

Parent elements
<a href="#">CategoryGrouping</a>

<b>Child elements</b>
<a href="#">DynamicCategories.Grouping</a>
<a href="#">DynamicCategories.Label</a>
<a href="#">DynamicCategories.Sorting</a>

The following is the XML Schema definition of the **DynamicCategories** element.

```
<xsd:complexType name="DynamicCategoriesType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Grouping" type="GroupingType" />
    <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.95.1 DynamicCategories.Grouping

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicCategories.Grouping** element specifies the expression by which to group the data. Page breaks in the grouping MUST NOT be used (the value of [Grouping.PageBreakAtEnd](#) and [Grouping.PageBreakAtStart](#) MUST be false). The **DynamicCategories.Grouping** element MUST be specified and is of type [Grouping](#).

Following is the parent element of the **DynamicCategories.Grouping** element.

<b>Parent elements</b>
<a href="#">DynamicCategories</a>

The following is the XML Schema definition of the **DynamicCategories.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" />
```

### 2.95.2 DynamicCategories.Label

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicCategories.Label** element specifies the label displayed on an [Axis](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **DynamicCategories.Label** element.

<b>Parent elements</b>
<a href="#">DynamicCategories</a>

The following is the XML Schema definition of the **DynamicCategories.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

### 2.95.3 DynamicCategories.Sorting

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicCategories.Sorting** element specifies the expressions by which to sort the data. This element is optional and is of type [Sorting](#).

Following is the parent element of the **DynamicCategories.Sorting** element.

Parent elements
<a href="#">DynamicCategories</a>

The following is the XML Schema definition of the **DynamicCategories.Sorting** element.

```
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
```

## 2.96 StaticCategories

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **StaticCategories** element specifies a category level with a fixed set of members in a [Chart](#). There **MUST** be one corresponding [DataPoint](#) for each **static member**.

The following are the parent and child elements of the **StaticCategories** element.

Parent elements
<a href="#">CategoryGrouping</a>

Child elements
<a href="#">StaticCategories.StaticMember</a>

The following is the XML Schema definition of the **StaticCategories** element.

```
<xsd:complexType name="StaticCategoriesType">
  <xsd:sequence>
    <xsd:element name="StaticMember" type="StaticMemberType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.96.1 StaticCategories.StaticMember

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **StaticCategories.StaticMember** element specifies a **member** of a **static category** level. This element **MUST** be specified and is of type [StaticMember](#).

Following is the parent element of the **StaticCategories.StaticMember** element.

Parent elements
<a href="#">StaticCategories</a>

The following is the XML Schema definition of the **StaticCategories.StaticMember** element.

```
<xsd:element name="StaticMember" type="StaticMemberType" maxOccurs="unbounded" />
```

## 2.97 StaticMember

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **StaticMember** element specifies the data label to display for a **static series** or **static category** member in a [Chart](#).

The following are the parent elements and the child element of the **StaticMember** element.

Parent elements
<a href="#">StaticCategories</a>
<a href="#">StaticSeries</a>

Child elements
<a href="#">StaticMember.Label</a>

The following is the XML Schema definition of the **StaticMember** element.

```
<xsd:complexType name="StaticMemberType">  
  <xsd:choice minOccurs="1" maxOccurs="unbounded">  
    <xsd:element name="Label" type="xsd:string" />  
    <xsd:any namespace="##other" processContents="skip" />  
  </xsd:choice>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

### 2.97.1 StaticMember.Label

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **StaticMember.Label** element specifies the label for a static member. This element **MUST** be specified and **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **StaticMember.Label** element.

Parent elements
<a href="#">StaticMember</a>

The following is the XML Schema definition of the **StaticMember.Label** element.

```
<xsd:element name="Label" type="xsd:string" />
```

## 2.98 ChartData

The **ChartData** element defines the structure and segmentation of data into multiple series. In [RDL 2003/10](#) and [RDL 2005/01](#), the **ChartData** element MUST contain at least one [ChartSeries](#) element. In [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#), the **ChartData** element MUST contain one [ChartSeriesCollection](#) element.

The following are the parent elements of the **ChartData** element.

Parent elements
<a href="#">Chart</a>

The following are the child elements of the **ChartData** element in RDL 2003/10 and RDL 2005/01.

Child elements
<a href="#">ChartData.ChartSeries</a>

The following are the child elements of the **ChartData** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Child elements
<a href="#">ChartData.ChartSeriesCollection</a>
<a href="#">ChartData.ChartDerivedSeriesCollection</a>

The following is the XML Schema definition of the **ChartData** element in RDL 2003/10 and RDL 2005/01.

```
<xsd:complexType name="ChartData">
  <xsd:sequence>
    <xsd:element name="ChartSeries" type="ChartSeriesType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartData** element in RDL 2008/01.

```
<xsd:complexType name="ChartData">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="ChartSeriesCollection" type="ChartSeriesCollectionType"
      minOccurs="1" maxOccurs="1" />
    <xsd:element name="ChartDerivedSeriesCollection"
      type="ChartDerivedSeriesCollectionType" minOccurs="0" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartData** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="ChartData" >
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="ChartSeriesCollection" type="ChartSeriesCollectionType"
      minOccurs="1" maxOccurs="1" />
    <xsd:element name="ChartDerivedSeriesCollection"
      type="ChartDerivedSeriesCollectionType" minOccurs="0" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.98.1 ChartData.ChartSeries

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **ChartData.ChartSeries** element specifies the ordered list of series in a [Chart](#). This element is of type [ChartSeries](#).

Following is the parent element of the **ChartData.ChartSeries** element.

<b>Parent elements</b>
<a href="#">ChartData</a>

The following is the XML Schema definition of the **ChartData.ChartSeries** element.

```

<xsd:element name="ChartSeries" type="ChartSeriesType" maxOccurs="unbounded" />

```

### 2.98.2 ChartData.ChartSeriesCollection

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartData.ChartSeriesCollection** element specifies data points for each series in a [Chart](#). The **ChartData.ChartSeriesCollection** element MUST be specified. This element MUST contain at least one [ChartSeries](#). This element is of type [ChartSeriesCollection](#).

Following is the parent element of the **ChartData.ChartSeriesCollection** element.

<b>Parent elements</b>
<a href="#">ChartData</a>

The following is the XML Schema definition of the **ChartData.ChartSeriesCollection** element.

```

<xsd:element name="ChartSeriesCollection" type="ChartSeriesCollectionType"
  minOccurs="1" maxOccurs="1" />

```

### 2.98.3 ChartData.ChartDerivedSeriesCollection

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartData.ChartDerivedSeriesCollection** element specifies the set of **derived series** for a [Chart](#). This element is optional. This element is of type [ChartDerivedSeriesCollection](#).

Following is the parent element of the **ChartData.ChartDerivedSeriesCollection** element.

<b>Parent elements</b>
<a href="#">ChartData</a>

The following is the XML Schema definition of the **ChartData.ChartDerivedSeriesCollection** element.

```
<xsd:element name="ChartDerivedSeriesCollection"
  type="ChartDerivedSeriesCollectionType" minOccurs="0" />
```

## 2.99 ChartSeries

The **ChartSeries** element specifies the **list of data points** for one chart series in [RDL 2003/10](#) and [RDL 2005/01](#), or for one series in a [ChartArea](#) in [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#).

In RDL 2008/01, RDL 2010/01, and RDL 2016/01, if there are multiple **ChartSeries** elements in a **ChartArea** and if these elements have values for [ChartSeries.Type](#) and/or [ChartSeries.Subtype](#) that cannot be displayed together in the same **ChartArea** (such as a column chart and a pie chart), then the values of the **ChartSeries.Type** and **ChartSeries.Subtype** of the first **ChartSeries** element in the [ChartSeriesCollection](#) are used to override the series that has an incompatible type or subtype. For example, if the first chart series type and/or subtype is a column chart, a pie chart is considered to be a column chart when it appears in the same **ChartArea** as the column chart.

The following are the parent elements of the **ChartSeries** element in RDL 2003/10 and RDL 2005/01.

<b>Parent elements</b>
<a href="#">ChartData</a>

The following are the parent elements of the **ChartSeries** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

<b>Parent elements</b>
ChartSeriesCollection
<a href="#">ChartDerivedSeries</a>

The following are the attributes of the **ChartSeries** element.

<b>Attributes</b>
<a href="#">ChartSeries.Name</a>

The following are the child elements of the **ChartSeries** element in RDL 2003/10 and RDL 2005/01.

<b>Child elements</b>
<a href="#">ChartSeries.DataPoints</a>
<a href="#">ChartSeries.PlotType</a>

The following are the child elements of the **ChartSeries** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.



Child elements
<a href="#">ChartSeries.CategoryAxisName</a>
<a href="#">ChartSeries.ChartAreaName</a>
<a href="#">ChartSeries.ChartDataLabel</a>
<a href="#">ChartSeries.ChartDataPoints</a>
<a href="#">ChartSeries.ChartEmptyPoints</a>
<a href="#">ChartSeries.ChartItemInLegend</a>
<a href="#">ChartSeries.ChartMarker</a>
<a href="#">ChartSeries.ChartSmartLabel</a>
<a href="#">ChartSeries.CustomProperties</a>
<a href="#">ChartSeries.Hidden</a>
<a href="#">ChartSeries.LegendName</a>
<a href="#">ChartSeries.Style</a>
ChartSeries.Subtype
ChartSeries.Type
<a href="#">ChartSeries.ValueAxisName</a>

The following is the XML Schema definition of the **ChartSeries** element in RDL 2003/10 and RDL 2005/01.

```
<xsd:complexType name="ChartSeriesType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataPoints" type="DataPointsType" />
    <xsd:element name="PlotType" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Auto" />
          <xsd:enumeration value="Line" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartSeries** element in RDL 2008/01.

```
<xsd:complexType name="ChartSeriesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartDataPoints" type="ChartDataPointsType"
      minOccurs="0" />
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <xsd:element name="Subtype" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartEmptyPoints" type="ChartEmptyPointsType"
      minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" />
  </xsd:choice>
</xsd:complexType>
```

```

        minOccurs="0" />
<xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
<xsd:element name="ChartItemInLegend" type="ChartItemInLegendType"
minOccurs="0" />
<xsd:element name="ChartAreaName" type="xsd:string" minOccurs="0" />
<xsd:element name="ValueAxisName" type="xsd:string" minOccurs="0" />
<xsd:element name="CategoryAxisName" type="xsd:string" minOccurs="0" />
<xsd:element name="ChartSmartLabel" type="ChartSmartLabelType"
minOccurs="0" />
<xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
<xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **ChartSeries** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="ChartSeriesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartDataPoints" type="ChartDataPointsType"
minOccurs="0" />
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <xsd:element name="Subtype" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartEmptyPoints" type="ChartEmptyPointsType"
minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartItemInLegend" type="ChartItemInLegendType"
minOccurs="0" />
    <xsd:element name="ChartAreaName" type="xsd:string" minOccurs="0" />
    <xsd:element name="ValueAxisName" type="xsd:string" minOccurs="0" />
    <xsd:element name="CategoryAxisName" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartSmartLabel" type="ChartSmartLabelType"
minOccurs="0" />
    <xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
    <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.99.1 ChartSeries.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSeries.Name** attribute specifies the name of a chart series. This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique among the **ChartSeries.Name** values in the parent collection.

Following is the parent element of the **ChartSeries.Name** attribute.

Parent elements
<a href="#">ChartSeries</a>

The following is the XML Schema definition of the **ChartSeries.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.99.2 ChartSeries.CategoryAxisName

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartSeries.CategoryAxisName** element specifies the name of the category axis against which to plot a [ChartSeries](#). This element is optional.

If the **ChartSeries.CategoryAxisName** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is the name of a [ChartAxis](#) element in the [ChartCategoryAxes](#) element. If the **ChartSeries.CategoryAxisName** element is not present, the series MUST be plotted against the first category axis. Following is the parent element of the **ChartSeries.CategoryAxisName** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.CategoryAxisName** element.

```
<xsd:element name="CategoryAxisName" type="xsd:string" minOccurs="0" />
```

### 2.99.3 ChartSeries.ChartAreaName

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartSeries.ChartAreaName** element specifies the name of the [ChartArea](#) in which to plot a [ChartSeries](#). The **ChartSeries.ChartAreaName** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is the name of a **ChartArea** in the [Chart](#). If this element is not present, the series MUST be plotted to the first chart area.

Following is the parent element of the **ChartSeries.ChartAreaName** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.ChartAreaName** element.

```
<xsd:element name="ChartAreaName" type="xsd:string" minOccurs="0" />
```

### 2.99.4 ChartSeries.ChartDataLabel

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartSeries.ChartDataLabel** element specifies the values in a [ChartSeries](#) that MUST be marked with data labels. This element is optional. This element is ignored if the parent element of the **ChartSeries** element is not [ChartDerivedSeries](#).

The **ChartSeries.ChartDataLabel** element is of type [ChartDataLabel](#).

Following is the parent element of the **ChartSeries.ChartDataLabel** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.ChartDataLabel** element.

```
<xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
```

### 2.99.5 ChartSeries.ChartDataPoints

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSeries.ChartDataPoints** element specifies the **data points** within a chart series. This element is optional.

If the parent of the [ChartSeries](#) element is [ChartData.ChartSeriesCollection](#), the **ChartSeries.ChartDataPoints** element MUST be specified. If the parent of the **ChartSeries** element is [ChartDerivedSeries](#), the **ChartSeries.ChartDataPoints** element is ignored.

This element is of type [ChartDataPoints](#).

Following is the parent element of the **ChartSeries.ChartDataPoints** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.ChartDataPoints** element.

```
<xsd:element name="ChartDataPoints" type="ChartDataPointsType" minOccurs="0" />
```

### 2.99.6 ChartSeries.ChartEmptyPoints

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSeries.ChartEmptyPoints** element specifies the behavior of **empty points** in a **ChartSeries**. This element is optional. This element is of type [ChartEmptyPoints](#).

Following is the parent element of the **ChartSeries.ChartEmptyPoints** element.

Parent elements
<a href="#">ChartSeries</a>

The following is the XML Schema definition of the **ChartSeries.ChartEmptyPoints** element.

```
<xsd:element name="ChartEmptyPoints" type="ChartEmptyPointsType" minOccurs="0" />
```

### 2.99.7 ChartSeries.ChartItemInLegend

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSeries.ChartItemInLegend** element specifies how a ChartSeries appears when it is displayed in a legend. This element is optional. This element is of type [ChartItemInLegend](#).

Following is the parent element of the **ChartSeries.ChartItemInLegend** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.ChartItemInLegend** element.

```
<xsd:element name="ChartItemInLegend" type="ChartItemInLegendType" minOccurs="0" />
```

## 2.99.8 ChartSeries.ChartMarker

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSeries.ChartMarker** element specifies the appearance of a data point marker. This element is optional. If the parent of the ChartSeries is not [ChartDerivedSeries](#), the **ChartSeries.ChartMarker** element is ignored.

This element is of type [ChartMarker](#).

Following is the parent element of the **ChartSeries.ChartMarker** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.ChartMarker** element.

```
<xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
```

## 2.99.9 ChartSeries.ChartSmartLabel

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSeries.ChartSmartLabel** element specifies the properties of a smart label. This element is optional. This element is of type [ChartSmartLabel](#).

Following is the parent element of the **ChartSeries.ChartSmartLabel** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.ChartSmartLabel** element.

```
<xsd:element name="ChartSmartLabel" type="ChartSmartLabelType" minOccurs="0" />
```

## 2.99.10 ChartSeries.CustomProperties

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSeries.CustomProperties** element specifies the custom properties for a ChartSeries. This element is optional. This element is of type [CustomProperties](#).

Following is the parent element of the **ChartSeries.CustomProperties** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

### 2.99.11 ChartSeries.DataPoints

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ChartSeries.DataPoints** element specifies a set of data points for a [ChartSeries](#). This element MUST be specified. The **ChartSeries.DataPoints** element is of type [DataPoints](#).

Following is the parent element of the **ChartSeries.DataPoints** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.DataPoints** element.

```
<xsd:element name="DataPoints" type="DataPointsType" />
```

### 2.99.12 ChartSeries.Hidden

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSeries.Hidden** element specifies whether a ChartSeries is hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartSeries.Hidden** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

### 2.99.13 ChartSeries.LegendName

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSeries.LegendName** element specifies the name of the [ChartLegend](#) in which a [ChartSeries](#) MUST appear. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1).

Following is the parent element of the **ChartSeries.LegendName** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.LegendName** element.

```
<xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
```

### 2.99.14 ChartSeries.PlotType

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ChartSeries.PlotType** element specifies whether a [ChartSeries](#) MUST be plotted as a line. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is one of the following:

**Auto** (Default): Specifies that whether to plot the series as a line is dependent on the values of the [Chart.Type](#) and [Chart.Subtype](#) elements.

**Line**: Specifies that the series is plotted as a line.

If the **ChartSeries.PlotType** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartSeries.PlotType** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.PlotType** element.

```
<xsd:element name="PlotType" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Auto" />
      <xsd:enumeration value="Line" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.99.15 ChartSeries.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSeries.Style** element specifies the style properties for a [ChartSeries](#). This element is optional. This element is of type [Style](#).

Following is the parent element of the **ChartSeries.Style** element.

<b>Parent elements</b>
ChartSeries

The following is the XML Schema definition of the **ChartSeries.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.99.16 ChartSeries.Subtype

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#), unless otherwise specified.

The **ChartSeries.Subtype** element specifies the visualization subtype for a ChartSeries. The available subtypes and the default subtype depend on the value of the [ChartSeries.Type](#) element. If an invalid subtype is specified, the default subtype for the specified type will be used.

The **ChartSeries.Subtype** element is optional.

If the **ChartSeries.Subtype** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/21](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Plain:** Specifies a plain visualization for all types.

**Stacked:** Specifies a stacked visualization for column, bar, and area charts. This value is not used if the value of the **ChartSeries.Type** element is not "Column", "Bar", or "Area".

**PercentStacked:** Specifies a column, bar, or area chart stacked by percentage. This value is not used if the value of the **ChartSeries.Type** element is not "Column", "Bar", or "Area".

**Smooth:** Specifies a smooth line, area, or range chart. This value is not used if the value of the **ChartSeries.Type** element is not "Line", "Area", or "Range".

**Stepped:** Specifies a stepped line chart. This value is not used if the value of the **ChartSeries.Type** element is not "Line".

**Pie:** Specifies a pie chart. This value is not used if the value of the **ChartSeries.Type** element is not "Shape".

**ExplodedPie:** Specifies an exploded pie chart. This value is not used if the value of the **ChartSeries.Type** element is not "Shape".

**Doughnut:** Specifies a doughnut chart. This value is not used if the value of the **ChartSeries.Type** element is not "Shape".

**Exploded Doughnut:** Specifies an exploded doughnut chart. This value is not used if the value of the **ChartSeries.Type** element is not "Shape".

**Funnel:** Specifies a funnel chart. This value is not used if the value of the **ChartSeries.Type** element is not "Shape".

**Pyramid:** Specifies a pyramid chart. This value is not used if the value of the **ChartSeries.Type** element is not "Shape".

**Bubble:** Specifies a bubble chart. This value is not used if the value of the **ChartSeries.Type** element is not "Scatter".

**Candlestick:** Specifies a candlestick chart. This value is not used if the value of the **ChartSeries.Type** element is not "Range".



**Stock:** Specifies a stock chart. This value is not used if the value of the **ChartSeries.Type** element is not "Range".

**Bar:** Specifies a range bar chart. This value is not used if the value of the **ChartSeries.Type** element is not "Range".

**Column:** Specifies a range column chart. This value is not used if the value of the **ChartSeries.Type** element is not "Range".

**BoxPlot:** Specifies a box plot chart. This value is not used if the value of the **ChartSeries.Type** element is not "Range".

**ErrorBar:** Specifies an error bar chart. This value is not used if the value of the **ChartSeries.Type** element is not "Range".

**Radar:** Specifies a radar chart. This value is not used if the value of the **ChartSeries.Type** element is not "Polar".

**Map:** In [RDL 2012/01](#), specifies a map chart. This value is not used if the value of the **ChartSeries.Type** element is not "Scatter".

**TreeMap:** In RDL 2016/01, specifies a TreeMap chart. This value is not used if the value of the **ChartSeries.Type** element is not "Shape".

**Sunburst:** In RDL 2016/01, specifies a Sunburst chart. This value is not used if the value of the **ChartSeries.Type** element is not "Shape".

If the **ChartSeries.Subtype** element is not present and if the value of the **ChartSeries.Type** element is "Shape", the value of the **ChartSeries.Subtype** element is interpreted as "Pie".

If the **ChartSeries.Subtype** element is not present and if the value of the **ChartSeries.Type** element is not "Shape", the value of the **ChartSeries.Subtype** element is interpreted as "Plain".

Following is the parent element of the **ChartSeries.Subtype** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.Subtype** element.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections [5.5](#), [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:element name="Subtype" type="xsd:string" minOccurs="0" />
```

## 2.99.17 ChartSeries.Type

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSeries.Type** element specifies the visualization type for a ChartSeries. This element is optional.

If the **ChartSeries.Type** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Column:** Specifies a column chart.

**Bar:** Specifies a bar chart.

**Line:** Specifies a line chart.

**Shape:** Specifies a shape chart.

**Scatter:** Specifies a scatter chart.

**Area:** Specifies an area chart.

**Range:** Specifies a range chart.

**Polar:** Specifies a polar chart.

If the **ChartSeries.Type** element is not present, its value is interpreted as "Column".

Following is the parent element of the **ChartSeries.Type** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.Type** element.

```
<xsd:element name="Type" type="xsd:string" minOccurs="0" />
```

### 2.99.18 ChartSeries.ValueAxisName

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSeries.ValueAxisName** element specifies the name of the value axis against which to plot a ChartSeries. The **ChartSeries.ValueAxisName** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is the name of a [ValueAxis](#) element in the [Chart](#). If the **ChartSeries.ValueAxisName** element is not present, the series MUST be plotted against the first **ValueAxis** element.

Following is the parent element of the **ChartSeries.ValueAxisName** element.

Parent elements
ChartSeries

The following is the XML Schema definition of the **ChartSeries.ValueAxisName** element.

```
<xsd:element name="ValueAxisName" type="xsd:string" minOccurs="0" />
```

### 2.100 DataPoints

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DataPoints** element specifies a collection of [DataPoint](#) instances for a [ChartSeries](#). The **DataPoints** element MUST be specified. There MUST be one corresponding **DataPoint** instance for each [StaticMember](#) within a static category.

The following are the parent and child elements of the **DataPoints** element.

<b>Parent elements</b>
<a href="#">ChartSeries</a>

<b>Child elements</b>
<a href="#">DataPoints.DataPoint</a>

The following is the XML Schema definition of the **DataPoints** element.

```
<xsd:complexType name="DataPointsType">
  <xsd:sequence>
    <xsd:element name="DataPoint" type="DataPointType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.100.1 DataPoints.DataPoint

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DataPoints.DataPoint** element specifies a collection of data values for a [DataPoint](#) in a [Chart](#). This element **MUST** be specified and is of type **DataPoint**.

Following is the parent element of the **DataPoints.DataPoint** element.

<b>Parent elements</b>
<a href="#">DataPoints</a>

The following is the XML Schema definition of the **DataPoints.DataPoint** element.

```
<xsd:element name="DataPoint" type="DataPointType" maxOccurs="unbounded" />
```

### 2.101 DataPoint

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DataPoint** element specifies a set of single data point for a [Chart](#). The **DataPoint** element can consist of a single-value expression or multiple-value expressions, depending on the values of the [Chart.Type](#) and [Chart.Subtype](#) elements.

The following are the parent and child elements of the **DataPoint** element.

<b>Parent elements</b>
<a href="#">DataPoints</a>

<b>Child elements</b>
<a href="#">DataPoint.Action</a>

Child elements
<a href="#">DataPoint.DataElementName</a>
<a href="#">DataPoint.DataElementOutput</a>
<a href="#">DataPoint.DataLabel</a>
<a href="#">DataPoint.DataValues</a>
<a href="#">DataPoint.Marker</a>
<a href="#">DataPoint.Style</a>

The following is the XML Schema definition of the **DataPoint** element.

```
<xsd:complexType name="DataPointType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataValues" type="DataValuesType" />
    <xsd:element name="DataLabel" type="DataLabelType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Marker" type="MarkerType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.101.1 DataPoint.Action

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DataPoint.Action** element specifies action that are associated with a [DataPoint](#) in a [Chart](#). The **DataPoint.Action** element is optional and is of type [Action](#).

Following is the parent element of the **DataPoint.Action** element.

Parent elements
DataPoint

The following is the XML Schema definition of the **DataPoint.Action** element.

```
<xsd:element name="Action" type="ActionType" minOccurs="0" />
```

### 2.101.2 DataPoint.DataElementName

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DataPoint.DataElementName** element specifies the name to use for the data element for a [DataPoint](#) in a [Chart](#). This element is optional. If the **DataPoint.DataElementName** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). If this element is not present, its value is interpreted as the name of the corresponding static series or static category.

Following is the parent element of the **DataPoint.DataElementName** element.

Parent elements
DataPoint

The following is the XML Schema definition of the **DataPoint.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.101.3 DataPoint.DataElementOutput

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **DataPoint.DataElementOutput** element specifies whether a [DataPoint](#) appears in a **data rendering**. This element is optional.

If the **DataPoint.DataElementOutput** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**Output** (default): Specifies that the data point appears in the data rendering output.

**NoOutput**: Specifies that the data point does not appear in the data rendering output.

If the **DataPoint.DataElementOutput** element is not present, its value is interpreted as "Output".

Following is the parent element of the **DataPoint.DataElementOutput** element.

Parent elements
DataPoint

The following is the XML Schema definition of the **DataPoint.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.101.4 DataPoint.DataLabel

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **DataPoint.DataLabel** element specifies the data label for a [DataPoint](#). This element is optional and is of type [DataLabel](#).

Following is the parent element of the **DataPoint.DataLabel** element.

Parent elements
DataPoint

The following is the XML Schema definition of the **DataPoint.DataLabel** element.

```
<xsd:element name="DataLabel" type="DataLabelType" minOccurs="0" />
```

### 2.101.5 DataPoint.DataValues

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DataPoint.DataValues** element specifies the set data values for a [DataPoint](#). This element **MUST** be specified and is of type [DataValues](#).

Following is the parent element of the **DataPoint.DataValues** element.

Parent elements
DataPoint

The following is the XML Schema definition of the **DataPoint.DataValues** element.

```
<xsd:element name="DataValues" type="DataValuesType" />
```

### 2.101.6 DataPoint.Marker

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DataPoint.Marker** element specifies the appearance of a data point marker. This element is optional. This element is ignored if the value of the [Chart.Type](#) element is "Pie" or "Doughnut" or if the value of the [Chart.Subtype](#) element is "Stacked" or "PercentStacked".

The **DataPoint.Marker** element is of type [Marker](#).

Following is the parent element of the **DataPoint.Marker** element.

Parent elements
<a href="#">DataPoint</a>

The following is the XML Schema definition of the **DataPoint.Marker** element.

```
<xsd:element name="Marker" type="MarkerType" minOccurs="0" />
```

### 2.101.7 DataPoint.Style

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DataPoint.Style** element specifies border and background style properties for a [DataPoint](#). This element is optional and is of type [Style](#).

Following is the parent element of the **DataPoint.Style** element.

Parent elements
DataPoint

The following is the XML Schema definition of the **DataPoint.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.102 DataLabel

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DataLabel** element specifies the data labels to display on data values in a [Chart](#).

The following are the parent and child elements of the **DataLabel** element.

Parent elements
<a href="#">DataPoint</a>

Child elements
<a href="#">DataLabel.Position</a>
<a href="#">DataLabel.Rotation</a>
<a href="#">DataLabel.Style</a>
<a href="#">DataLabel.Value</a>
<a href="#">DataLabel.Visible</a>

The following is the XML Schema definition of the **DataLabel** element.

```
<xsd:complexType name="DataLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Visible" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Auto" />
          <xsd:enumeration value="Top" />
          <xsd:enumeration value="TopLeft" />
          <xsd:enumeration value="TopRight" />
          <xsd:enumeration value="Left" />
          <xsd:enumeration value="Center" />
          <xsd:enumeration value="Right" />
          <xsd:enumeration value="BottomLeft" />
          <xsd:enumeration value="Bottom" />
          <xsd:enumeration value="BottomRight" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Rotation" type="xsd:integer" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</complexType>
```

</xsd:complexType>

### 2.102.1 DataLabel.Position

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DataLabel.Position** element specifies the position of a [DataLabel](#) in a [Chart](#). This element is optional. If this element is present, its value MUST be a [String](#) ([IXMLSCHEMA2/2](#) section 3.2.1) that is one of the following:

**Auto** (default): Specifies automatic positioning.

**Top**: Positions the data label at the top of the data point.

**TopLeft**: Positions the data label at the top-left of the data point.

**TopRight**: Positions the data label at the top-right of the data point.

**Left**: Positions the data label to the left of the data point.

**Center**: Positions the data label at the center of the data point.

**Right**: Positions the data label to the right of the data point.

**BottomRight**: Positions the data label at the bottom-right of the data point.

**Bottom**: Positions the data label at the bottom of the data point.

**BottomLeft**: Positions the data label at the bottom-left of the data point.

If the **DataLabel.Position** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **DataLabel.Position** element.

Parent elements
DataLabel

The following is the XML Schema definition of the **DataLabel.Position** element.

```
<xsd:element name="Position" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Auto" />
      <xsd:enumeration value="Top" />
      <xsd:enumeration value="TopLeft" />
      <xsd:enumeration value="TopRight" />
      <xsd:enumeration value="Left" />
      <xsd:enumeration value="Center" />
      <xsd:enumeration value="Right" />
      <xsd:enumeration value="BottomLeft" />
      <xsd:enumeration value="Bottom" />
      <xsd:enumeration value="BottomRight" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```



## 2.102.2 DataLabel.Rotation

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **DataLabel.Rotation** element specifies the angle of rotation for the text of a [DataLabel](#). This element is optional. If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, then its value is interpreted as 0.

Following is the parent element of the **DataLabel.Rotation** element.

Parent elements
<a href="#">DataLabel</a>

The following is the XML Schema definition of the **DataLabel.Rotation** element.

```
<xsd:element name="Rotation" type="xsd:integer" minOccurs="0" />
```

## 2.102.3 DataLabel.Style

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **DataLabel.Style** element specifies text, border, and background style properties for data labels. This element is optional and is of type [Style](#).

Following is the parent element of the **DataLabel.Style** element.

Parent elements
<a href="#">DataLabel</a>

The following is the XML Schema definition of the **DataLabel.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.102.4 DataLabel.Value

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **DataLabel.Value** element specifies the text for a [DataLabel](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **DataLabel.Value** element.

Parent elements
<a href="#">DataLabel</a>

The following is the XML Schema definition of the **DataLabel.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="0" />
```

## 2.102.5 DataLabel.Visible

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **DataLabel.Visible** element specifies whether this [DataLabel](#) appears on data values. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, then its value is interpreted as false.

The following is the parent element of the **DataLabel.Visible** element.

Parent elements
<a href="#">DataLabel</a>

The following is the XML Schema definition of the **DataLabel.Visible** element.

```
<xsd:element name="Visible" type="xsd:boolean" minOccurs="0" />
```

## 2.103 DataValues

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **DataValues** element specifies a set of [DataValue](#) instances for a [DataPoint](#) in a [Chart](#).

The following are the parent and child elements of the **DataValues** element.

Parent elements
<a href="#">DataPoint</a>

Child elements
<a href="#">DataValues.DataValue</a>

The following is the XML Schema definition of the **DataValues** element.

```
<xsd:complexType name="DataValuesType">
  <xsd:sequence>
    <xsd:element name="DataValue" type="DataValueType" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.103.1 DataValues.DataValue

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **DataValues.DataValue** element specifies a data value. This element MUST be specified and MUST contain at least one instance. This element is of type [DataValue](#).

Following is the parent element of the **DataValues.DataValue** element.

<b>Parent elements</b>
<a href="#">DataValues</a>

The following is the XML Schema definition of the **DataValues.DataValue** element.

```
<xsd:element name="DataValue" type="DataValueType" minOccurs="unbounded" />
```

## 2.104 Marker

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **Marker** element specifies a marker for displayed chart values.

The following are the parent and child elements of the **Marker** element.

<b>Parent elements</b>
<a href="#">DataPoint</a>

<b>Child elements</b>
<a href="#">Marker.Size</a>
<a href="#">Marker.Style</a>
<a href="#">Marker.Type</a>

The following is the XML Schema definition of the **Marker** element.

```
<xsd:complexType name="MarkerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Type" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Square" />
          <xsd:enumeration value="Circle" />
          <xsd:enumeration value="Diamond" />
          <xsd:enumeration value="Triangle" />
          <xsd:enumeration value="Cross" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Size" type="SizeType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.104.1 Marker.Size

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The **Marker.Size** element specifies the height and width of the plotting area of markers for chart values. This element is optional. If this element is present, its value MUST be an [RdISize](#) or an expression that evaluates to an **RdISize**. If this element is not present, the marker will not appear.

Following is the parent element of the **Marker.Size** element.

Parent elements
<a href="#">Marker</a>

The following is the XML Schema definition of the **Marker.Size** element.

```
<xsd:element name="Size" type="SizeType" minOccurs="0" />
```

## 2.104.2 Marker.Style

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Marker.Style** element specifies the border and background style properties for markers of chart values. This element is optional and is of type [Style](#).

Following is the parent element of the **Marker.Style** element.

Parent elements
<a href="#">Marker</a>

The following is the XML Schema definition of the **Marker.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.104.3 Marker.Type

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Marker.Type** element specifies the type of the markers for chart values. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) and MUST be one of the following:

**None** (default): Specifies that the marker is not displayed.

**Square**: Specifies a square marker.

**Circle**: Specifies a circle marker.

**Diamond**: Specifies a diamond marker.

**Triangle**: Specifies a triangle marker.

**Cross**: Specifies a cross marker.

**Auto**: Specifies to automatically cycle through marker types for each series.

If the **Marker.Type** element is not present, its value is interpreted as "None".

Following is the parent element of the **Marker.Type** element.

<b>Parent elements</b>
------------------------

<a href="#">Marker</a>
------------------------

The following is the XML Schema definition of the **Marker.Type** element.

```
<xsd:element name="Type" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None" />
      <xsd:enumeration value="Square" />
      <xsd:enumeration value="Circle" />
      <xsd:enumeration value="Diamond" />
      <xsd:enumeration value="Triangle" />
      <xsd:enumeration value="Cross" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.105 Legend

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Legend** element specifies the properties that can be used to display instances of the series groups in a chart legend.

The following are the parent and child elements of the **Legend** element.

<b>Parent elements</b>
------------------------

<a href="#">Chart</a>
-----------------------

<b>Child elements</b>
-----------------------

<a href="#">Legend.InsidePlotArea</a>
---------------------------------------

<a href="#">Legend.Layout</a>
-------------------------------

<a href="#">Legend.Position</a>
---------------------------------

<a href="#">Legend.Visible</a>
--------------------------------

The following is the XML Schema definition of the **Legend** element.

```
<xsd:complexType name="LegendType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Visible" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Position" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="TopLeft" />
          <xsd:enumeration value="TopCenter" />
          <xsd:enumeration value="TopRight" />
          <xsd:enumeration value="LeftTop" />
          <xsd:enumeration value="LeftCenter" />
          <xsd:enumeration value="LeftBottom" />
          <xsd:enumeration value="RightTop" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>
```

```

        <xsd:enumeration value="RightCenter" />
        <xsd:enumeration value="RightBottom" />
        <xsd:enumeration value="BottomLeft" />
        <xsd:enumeration value="BottomCenter" />
        <xsd:enumeration value="BottomRight" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="Layout" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Column" />
            <xsd:enumeration value="Row" />
            <xsd:enumeration value="Table" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="InsidePlotArea" type="xsd:boolean" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.105.1 Legend.InsidePlotArea

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Legend.InsidePlotArea** element specifies whether a legend is drawn inside a [PlotArea](#). This element is optional. If the **Legend.InsidePlotArea** element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, the value of this element is interpreted as false, indicating that the legend will be drawn outside of the **plot area**.

Following is the parent element of the **Legend.InsidePlotArea** element.

Parent elements
<a href="#">Legend</a>

The following is the XML Schema definition of the **Legend.InsidePlotArea** element.

```
<xsd:element name="InsidePlotArea" type="xsd:boolean" minOccurs="0" />
```

### 2.105.2 Legend.Layout

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Legend.Layout** element specifies the arrangement of labels within a legend for a series group. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**Column** (default): Specifies that labels are arranged in a column.

**Row**: Specifies that labels are arranged in a row.

**Table**: Specifies that labels are arranged in a table.

If the **Legend.Layout** element is not present, its value is interpreted as "Column".

Following is the parent element of the **Legend.Layout** element.

<b>Parent elements</b>
------------------------

<a href="#">Legend</a>
------------------------

The following is the XML Schema definition of the **Legend.Layout** element.

```
<xsd:element name="Layout" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Column" />
      <xsd:enumeration value="Row" />
      <xsd:enumeration value="Table" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.105.3 Legend.Position

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Legend.Position** element specifies the position of the chart legend. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**RightTop:** Positions the legend at the upper corner of the right side of the chart.

**TopLeft:** Positions the legend at the left corner of the upper side of the chart.

**TopCenter:** Positions the legend at the middle of the upper side of the chart.

**TopRight:** Positions the legend at the right corner of the upper side of the chart.

**LeftTop:** Positions the legend at the upper corner of the left side of the chart.

**LeftCenter:** Positions the legend at the middle of the left side of the chart.

**LeftBottom:** Positions the legend at the bottom corner of the left side of the chart.

**RightCenter:** Positions the legend at the middle of the right side of the chart.

**RightBottom:** Positions the legend at the bottom corner of the right side of the chart.

**BottomRight:** Positions the legend at the right corner of the bottom side of the chart.

**BottomCenter:** Positions the legend at the middle of the bottom side of the chart.

**BottomLeft:** Positions the legend at the left corner of the bottom side of the chart.

If the **Legend.Position** element is not present, its value is interpreted as "RightTop".

Following is the parent element of the **Legend.Position** element.

<b>Parent elements</b>
------------------------

<a href="#">Legend</a>
------------------------

The following is the XML Schema definition of the **Legend.Position** element.

```
<xsd:element name="Position" minOccurs="0">
  <xsd:simpleType>
```

```

<xsd:restriction base="xsd:string">
  <xsd:enumeration value="TopLeft" />
  <xsd:enumeration value="TopCenter" />
  <xsd:enumeration value="TopRight" />
  <xsd:enumeration value="LeftTop" />
  <xsd:enumeration value="LeftCenter" />
  <xsd:enumeration value="LeftBottom" />
  <xsd:enumeration value="RightTop" />
  <xsd:enumeration value="RightCenter" />
  <xsd:enumeration value="RightBottom" />
  <xsd:enumeration value="BottomLeft" />
  <xsd:enumeration value="BottomCenter" />
  <xsd:enumeration value="BottomRight" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>

```

## 2.105.4 Legend.Visible

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Legend.Visible** element specifies whether a legend is displayed. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, the value of this element is interpreted as false.

Following is the parent element of the **Legend.Visible** element.

Parent elements
<a href="#">Legend</a>

The following is the XML Schema definition of the **Legend.Visible** element.

```
<xsd:element name="Visible" type="xsd:boolean" minOccurs="0" />
```

## 2.106 PlotArea

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **PlotArea** element specifies the properties for a **plot area**.

The following are the parent and child elements of the **PlotArea** element.

Parent elements
<a href="#">Chart</a>

Child elements
<a href="#">PlotArea.Style</a>

The following is the XML Schema definition of the **PlotArea** element.

```

<xsd:complexType name="PlotAreaType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />

```



```

    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.106.1 PlotArea.Style

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **PlotArea.Style** element specifies borders and background for a **plot area**. This element is optional and is of type [Style](#).

Following is the parent element of the **PlotArea.Style** element.

Parent elements
<a href="#">PlotArea</a>

The following is the XML Schema definition of the **PlotArea.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.107 SeriesGroupings

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **SeriesGroupings** element specifies the set of [SeriesGrouping](#) instances for a [Chart](#).

The following are the parent and child elements of the **SeriesGroupings** element.

Parent elements
Chart

Child elements
<a href="#">SeriesGroupings.SeriesGrouping</a>

The following is the XML Schema definition of the **SeriesGroupings** element.

```

<xsd:complexType name="SeriesGroupingsType">
  <xsd:sequence>
    <xsd:element name="SeriesGrouping" type="SeriesGroupingType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

#### 2.107.1 SeriesGroupings.SeriesGrouping

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **SeriesGroupings.SeriesGrouping** element specifies a series grouping instance. This element is of type [SeriesGrouping](#).

Following is the parent element of the **SeriesGroupings.SeriesGrouping** element.

Parent elements
<a href="#">SeriesGroupings</a>

The following is the XML Schema definition of the **SeriesGroupings.SeriesGrouping** element.

```
<xsd:element name="SeriesGrouping" type="SeriesGroupingType"
maxOccurs="unbounded" />
```

## 2.108 SeriesGrouping

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **SeriesGrouping** element specifies a series level for a [Chart](#). The labels for the series are displayed in the legend. The **SeriesGrouping** element MUST have either a **dynamic series** or a **static series**, but not both.

The following are the parent and child elements of the **SeriesGrouping** element.

Parent elements
<a href="#">SeriesGroupings</a>

Child elements
<a href="#">SeriesGrouping.DynamicSeries</a>
<a href="#">SeriesGrouping.StaticSeries</a>

The following is the XML Schema definition of the **SeriesGrouping** element.

```
<xsd:complexType name="SeriesGroupingType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DynamicSeries" type="DynamicSeriesType" minOccurs="0" />
    <xsd:element name="StaticSeries" type="StaticSeriesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.108.1 SeriesGrouping.DynamicSeries

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **SeriesGrouping.DynamicSeries** element specifies the collection of **dynamic series** headings for a grouping. This element is optional and is of type [DynamicSeries](#).

Following is the parent element of the **SeriesGrouping.DynamicSeries** element.

<b>Parent elements</b>
------------------------

<a href="#">SeriesGrouping</a>
--------------------------------

The following is the XML Schema definition of the **SeriesGrouping.DynamicSeries** element.

```
<xsd:element name="DynamicSeries" type="DynamicSeriesType" minOccurs="0" />
```

## 2.108.2 SeriesGrouping.StaticSeries

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **SeriesGrouping.StaticSeries** element specifies the collection of **static series** headings for a grouping. This element is optional and is of type [StaticSeries](#).

Following is the parent element of the **SeriesGrouping.StaticSeries** element.

<b>Parent elements</b>
------------------------

<a href="#">SeriesGrouping</a>
--------------------------------

The following is the XML Schema definition of the **SeriesGrouping.StaticSeries** element.

```
<xsd:element name="StaticSeries" type="StaticSeriesType" minOccurs="0" />
```

## 2.109 DynamicSeries

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **DynamicSeries** element specifies a series level that repeats with each series group in a rowset of a [Chart](#).

The following are the parent and child elements of the **DynamicSeries** element.

<b>Parent elements</b>
------------------------

<a href="#">SeriesGrouping</a>
--------------------------------

<b>Child elements</b>
-----------------------

<a href="#">DynamicSeries.Grouping</a>
--

<a href="#">DynamicSeries.Label</a>
-------------------------------------

<a href="#">DynamicSeries.Sorting</a>
---------------------------------------

The following is the XML Schema definition of the **DynamicSeries** element.

```
<xsd:complexType name="DynamicSeriesType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Grouping" type="GroupingType" />
    <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
</xsd:complexType>
```

```
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

## 2.109.1 DynamicSeries.Grouping

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicSeries.Grouping** element specifies the expression by which to group the data. Page breaks in the grouping MUST NOT be used (the value of [Grouping.PageBreakAtEnd](#) and [Grouping.PageBreakAtStart](#) MUST be false). The **DynamicSeries.Grouping** element MUST be specified and is of type [Grouping](#).

Following is the parent element of the **DynamicSeries.Grouping** element.

Parent elements
<a href="#">DynamicSeries</a>

The following is the XML Schema definition of the **DynamicSeries.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" />
```

## 2.109.2 DynamicSeries.Label

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicSeries.Label** element specifies the label that is displayed on a [Legend](#). This element MUST be specified. The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **DynamicSeries.Label** element.

Parent elements
<a href="#">DynamicSeries</a>

The following is the XML Schema definition of the **DynamicSeries.Label** element.

```
<xsd:element name="Label" type="xsd:string" />
```

## 2.109.3 DynamicSeries.Sorting

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **DynamicSeries.Sorting** element specifies the expressions by which to sort columns in a **dynamic series**. This element is optional and is of type [Sorting](#).

Following is the parent element of the **DynamicSeries.Sorting** element.

Parent elements
<a href="#">DynamicSeries</a>

The following is the XML Schema definition of the **DynamicSeries.Sorting** element.

```
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
```

## 2.110 StaticSeries

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **StaticSeries** element specifies a series level that has a fixed set of members.

The following are the parent and child elements of the **StaticSeries** element.

Parent elements
<a href="#">SeriesGrouping</a>

Child elements
<a href="#">StaticSeries.StaticMember</a>

The following is the XML Schema definition of the **StaticSeries** element.

```
<xsd:complexType name="StaticSeriesType">
  <xsd:sequence>
    <xsd:element name="StaticMember" type="StaticMemberType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.110.1 StaticSeries.StaticMember

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **StaticSeries.StaticMember** element specifies a member of a **static series**. This element **MUST** be specified and is of type [StaticMember](#).

Following is the parent element of the **StaticSeries.StaticMember** element.

Parent elements
<a href="#">StaticSeries</a>

The following is the XML Schema definition of the **StaticSeries.StaticMember** element.

```
<xsd:element name="StaticMember" type="StaticMemberType" maxOccurs="unbounded" />
```

## 2.111 ThreeDProperties

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ThreeDProperties** element specifies the properties of a 3D layout of a [Chart](#).

The following are the parent and child elements of the **ThreeDProperties** element.

Parent elements
Chart

Child elements
<a href="#">ThreeDProperties.Clustered</a>
<a href="#">ThreeDProperties.DepthRatio</a>
<a href="#">ThreeDProperties.DrawingStyle</a>
<a href="#">ThreeDProperties.Enabled</a>
<a href="#">ThreeDProperties.GapDepth</a>
<a href="#">ThreeDProperties.HeightRatio</a>
<a href="#">ThreeDProperties.Inclination</a>
<a href="#">ThreeDProperties.Perspective</a>
<a href="#">ThreeDProperties.ProjectionMode</a>
<a href="#">ThreeDProperties.Rotation</a>
<a href="#">ThreeDProperties.Shading</a>
<a href="#">ThreeDProperties.WallThickness</a>

The following is the XML Schema definition of the **ThreeDProperties** element.

```
<xsd:complexType name="ThreeDPropertiesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Enabled" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ProjectionMode" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Perspective" />
          <xsd:enumeration value="Orthographic" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Rotation" type="xsd:integer" minOccurs="0" />
    <xsd:element name="Inclination" type="xsd:integer" minOccurs="0" />
    <xsd:element name="Perspective" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="HeightRatio" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="DepthRatio" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Shading" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Simple" />
          <xsd:enumeration value="Real" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GapDepth" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="WallThickness" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="DrawingStyle" minOccurs="0">
      <xsd:simpleType>
```

```

    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Cube" />
      <xsd:enumeration value="Cylinder" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="Clustered" type="xsd:boolean" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.111.1 ThreeDProperties.Clustered

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ThreeDProperties.Clustered** element specifies whether data series in a [Chart](#) are clustered (displayed along distinct rows). This element is optional. Data series MUST NOT be displayed as clustered if the value of [Chart.Type](#) is not "Bar" or "Column". If the **ThreeDProperties.Clustered** element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **ThreeDProperties.Clustered** element.

<b>Parent elements</b>
<a href="#">ThreeDProperties</a>

The following is the XML Schema definition of the **ThreeDProperties.Clustered** element.

```
<xsd:element name="Clustered" type="xsd:boolean" minOccurs="0" />
```

### 2.111.2 ThreeDProperties.DepthRatio

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ThreeDProperties.DepthRatio** element MUST NOT be used.

Following is the parent element of the **ThreeDProperties.DepthRatio** element.

<b>Parent elements</b>
<a href="#">ThreeDProperties</a>

The following is the XML Schema definition of the **ThreeDProperties.DepthRatio** element.

```
<xsd:element name="DepthRatio" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.111.3 ThreeDProperties.DrawingStyle

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ThreeDProperties.DrawingStyle** element specifies the shape of column and bars in a column or bar [Chart](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**Cylinder:** Specifies that columns and bars are displayed as a cylinder.

**Cube** (default): Specifies that columns and bars are displayed as a cube.

If the **ThreeDProperties.DrawingStyle** element is not present, its value is interpreted as "Cube". If the value of the [Chart.Type](#) element is not "Bar" or "Column", the **ThreeDProperties.DrawingStyle** element is ignored.

Following is the parent element of the **ThreeDProperties.DrawingStyle** element.

<b>Parent elements</b>
<a href="#">ThreeDProperties</a>

The following is the XML Schema definition of the **ThreeDProperties.DrawingStyle** element.

```
<xsd:element name="DrawingStyle" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Cube" />
      <xsd:enumeration value="Cylinder" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

#### 2.111.4 ThreeDProperties.Enabled

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ThreeDProperties.Enabled** element specifies whether a [Chart](#) is displayed in 3D. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **ThreeDProperties.Enabled** element.

<b>Parent elements</b>
<a href="#">ThreeDProperties</a>

The following is the XML Schema definition of the **ThreeDProperties.Enabled** element.

```
<xsd:element name="Enabled" type="xsd:boolean" minOccurs="0" />
```

#### 2.111.5 ThreeDProperties.GapDepth

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ThreeDProperties.GapDepth** element is ignored if it is present. However, its data type is validated. Its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) and MUST NOT be less than 0.

Following is the parent element of the **ThreeDProperties.GapDepth** element.

<b>Parent elements</b>
<a href="#">ThreeDProperties</a>



The following is the XML Schema definition of the **ThreeDProperties.GapDepth** element.

```
<xsd:element name="GapDepth" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.111.6 ThreeDProperties.HeightRatio

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ThreeDProperties.HeightRatio** element is ignored if it is present. However, its data type is validated. Its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) and MUST NOT be less than 0.

The following is the parent element of the **ThreeDProperties.HeightRatio** element.

Parent elements
<a href="#">ThreeDProperties</a>

The following is the XML Schema definition of the **ThreeDProperties.HeightRatio** element.

```
<xsd:element name="HeightRatio" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.111.7 ThreeDProperties.Inclination

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ThreeDProperties.Inclination** element specifies an inclination angle. This element is optional. If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17). If this element is not present, its value is interpreted as 0. The value of this element MUST be greater than or equal to -90 and less than or equal to 90.

Following is the parent element of the **ThreeDProperties.Inclination** element.

Parent elements
<a href="#">ThreeDProperties</a>

The following is the XML Schema definition of the **ThreeDProperties.Inclination** element.

```
<xsd:element name="Inclination" type="xsd:integer" minOccurs="0" />
```

### 2.111.8 ThreeDProperties.Perspective

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ThreeDProperties.Perspective** element specifies the percentage of perspective in a 3D [Chart](#). This element is optional. This element is ignored if the value of [ThreeDProperties.ProjectionMode](#) is not "Perspective".

If the **ThreeDProperties.Perspective** element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17). If this element is not present, its value is interpreted as 0. The value of the **ThreeDProperties.Perspective** element MUST be greater than or equal to 0 and less than or equal to 100.

The following is the parent element of the **ThreeDProperties.Perspective** element.

Parent elements
<a href="#">ThreeDProperties</a>

The following is the XML Schema definition of the **ThreeDProperties.Perspective** element.

```
<xsd:element name="Perspective" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.111.9 ThreeDProperties.ProjectionMode

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ThreeDProperties.ProjectionMode** element specifies the projection mode used for a 3D rendering. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**Perspective** (default): Specifies that perspective projection is used.

**Orthographic**: Specifies that orthographic projection is used.

If the **ThreeDProperties.ProjectionMode** element is not present, its value is interpreted as "Perspective".

Following is the parent element of the **ThreeDProperties.ProjectionMode** element.

Parent elements
<a href="#">ThreeDProperties</a>

The following is the XML Schema definition of the **ThreeDProperties.ProjectionMode** element.

```
<xsd:element name="ProjectionMode" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Perspective" />
      <xsd:enumeration value="Orthographic" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.111.10 ThreeDProperties.Rotation

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ThreeDProperties.Rotation** element specifies a rotation angle of a 3D [Chart](#). This element is optional.

If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17). If this element is not present, its value is interpreted as 0. The value of this element MUST be greater than or equal to -90 and less than or equal to 90.

Following is the parent element of the **ThreeDProperties.Rotation** element.

<b>Parent elements</b>
------------------------

<a href="#">ThreeDProperties</a>
----------------------------------

The following is the XML Schema definition of the **ThreeDProperties.Rotation** element.

```
<xsd:element name="Rotation" type="xsd:integer" minOccurs="0" />
```

### 2.111.11 ThreeDProperties.Shading

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ThreeDProperties.Shading** element specifies the type of 3D shading in a 3D [Chart](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**Real:** Specifies realistic shading.

**Simple:** Specifies simplified shading.

**None** (default): Specifies no shading.

If this element is not present, its value is interpreted as "None".

The following is the parent element of the **ThreeDProperties.Shading** element.

<b>Parent elements</b>
------------------------

<a href="#">ThreeDProperties</a>
----------------------------------

The following is the XML Schema definition of the **ThreeDProperties.Shading** element.

```
<xsd:element name="Shading" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None" />
      <xsd:enumeration value="Simple" />
      <xsd:enumeration value="Real" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.111.12 ThreeDProperties.WallThickness

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ThreeDProperties.WallThickness** element specifies the percent thickness of outer walls in a 3D [Chart](#). This element is optional.

If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17). If this element is not present, its value is interpreted as 0. The value of this element MUST greater than or equal to 0 and less than or equal to 100.

Following is the parent element of the **ThreeDProperties.WallThickness** element.

<b>Parent elements</b>
<a href="#">ThreeDProperties</a>

The following is the XML Schema definition of the **ThreeDProperties.WallThickness** element.

```
<xsd:element name="WallThickness" type="xsd:unsignedInt" minOccurs="0" />
```

## 2.112 Title

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Title** element specifies a title for a [Chart](#) or for an [Axis](#).

The following are the parent and child elements of the **Title** element.

<b>Parent elements</b>
<a href="#">Chart</a>
<a href="#">Axis</a>

<b>Child elements</b>
<a href="#">Title.Caption</a>
<a href="#">Title.Position</a>
<a href="#">Title.Style</a>

The following is the XML Schema definition of the **Title** element.

```
<xsd:complexType name="TitleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Caption" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Position" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Center" />
          <xsd:enumeration value="Near" />
          <xsd:enumeration value="Far" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.112.1 Title.Caption

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Title.Caption** element specifies the caption of the title for a [Chart](#) or for an [Axis](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an

expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **Title.Caption** element.

Parent elements
<a href="#">Title</a>

The following is the XML Schema definition of the **Title.Caption** element.

```
<xsd:element name="Caption" type="xsd:string" minOccurs="0" />
```

### 2.112.2 Title.Position

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Title.Position** element specifies the position of the title for a [Chart](#) or for an [Axis](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**Center** (default): Specifies that the title is positioned to the center of the chart or axis.

**Near**: Specifies that the title is positioned to the left of the chart or axis.

**Far**: Specifies that the title is positioned to the right of the chart or axis.

If the **Title.Position** element is not present, its value is interpreted as "Center". This element is ignored if this is a chart title.

Following is the parent element of the **Title.Position** element.

Parent elements
<a href="#">Title</a>

The following is the XML Schema definition of the **Title.Position** element.

```
<xsd:element name="Position" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Center" />
      <xsd:enumeration value="Near" />
      <xsd:enumeration value="Far" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.112.3 Title.Style

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Title.Style** element specifies text, border, and background style properties for the title for a [Chart](#) or for an [Axis](#). This element is optional and is of type [Style](#).

Following is the parent element of the **Title.Style** element.

<b>Parent elements</b>
<a href="#">Title</a>

The following is the XML Schema definition of the **Title.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.113 ValueAxis

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ValueAxis** element specifies a value axis for a [Chart](#).

The following are the parent and child elements of the **ValueAxis** element.

<b>Parent elements</b>
<a href="#">Chart</a>

<b>Child elements</b>
<a href="#">ValueAxis.Axis</a>

The following is the XML Schema definition of the **ValueAxis** element.

```
<xsd:complexType name="ValueAxisType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Axis" type="AxisType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

#### 2.113.1 ValueAxis.Axis

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **ValueAxis.Axis** element specifies the value axis for a [Chart](#). This element is optional and is of type [Axis](#).

Following is the parent element of the **ValueAxis.Axis** element.

<b>Parent elements</b>
<a href="#">ValueAxis</a>

The following is the XML Schema definition of the **ValueAxis.Axis** element.

```
<xsd:element name="Axis" type="AxisType" minOccurs="0" />
```

## 2.114 ChartAnnotations

*Applies to [RDL 2008/01](#)*

The **ChartAnnotations** element is ignored.

Following are the parent and child elements of the **ChartAnnotations** element.

Parent elements
<a href="#">Chart</a>

Child elements
<a href="#">ChartAnnotations.ChartAnnotation</a>

The following is the XML Schema definition of the **ChartAnnotations** element.

```
<xsd:complexType name="ChartAnnotationType">
  <xsd:sequence>
    <xsd:element name="ChartAnnotation" type="ChartAnnotationType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

### 2.114.1 ChartAnnotations.ChartAnnotation

*Applies to [RDL 2008/01](#)*

The **ChartAnnotations.ChartAnnotation** element is ignored.

Following is the parent element of the **ChartAnnotations.ChartAnnotation** element.

Parent elements
<a href="#">ChartAnnotations</a>

The following is the XML Schema definition of the **ChartAnnotations.ChartAnnotation** element.

```
<xsd:element name="ChartAnnotation" type="ChartAnnotationType" maxOccurs="unbounded" />
```

## 2.115 ChartAnnotation

*Applies to [RDL 2008/01](#)*

The **ChartAnnotation** element is ignored.

Following is the parent element of the **ChartAnnotation** element.

Parent elements
<a href="#">ChartAnnotations</a>

The following is the XML Schema definition of the **ChartAnnotation** element.

```

<xsd:complexType name="ChartAnnotationType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    </xsd:choice>
  </xsd:complexType>

```

## 2.116 ChartAreas

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAreas** element specifies the set of [ChartArea](#) instances for a [Chart](#). This element **MUST** contain at least one **ChartArea** instance.

The following are the parent and child elements of the **ChartAreas** element.

Parent elements
Chart

Child elements
<a href="#">ChartAreas.ChartArea</a>

The following is the XML Schema definition of the **ChartAreas** element in RDL 2008/01.

```

<xsd:complexType name="ChartAreasType">
  <xsd:sequence>
    <xsd:element name="ChartArea" type="ChartAreaType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **ChartAreas** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="ChartAreasType">
  <xsd:sequence>
    <xsd:element name="ChartArea" type="ChartAreaType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.116.1 ChartAreas.ChartArea

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAreas.ChartArea** element specifies a [ChartArea](#) within the [ChartAreas](#) of a [Chart](#). At least one instance of the **ChartAreas.ChartArea** element **MUST** be specified. This element is of type **ChartArea**.

Following is the parent element of the **ChartAreas.ChartArea** element.

Parent elements
ChartAreas



The following is the XML Schema definition of the **ChartAreas.ChartArea** element.

```
<xsd:element name="ChartArea" type="ChartAreaType" maxOccurs="unbounded" />
```

## 2.117 ChartArea

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartArea** element specifies a [Chart](#) to be drawn in a **chart data region**.

Following are the parent elements, attributes, and child elements of the **ChartArea** element.

Parent elements
<a href="#">ChartAreas</a>

Attributes
<a href="#">ChartArea.Name</a>

Child elements
<a href="#">ChartArea.AlignOrientation</a>
<a href="#">ChartArea.AlignWithChartArea</a>
<a href="#">ChartArea.ChartAlignType</a>
<a href="#">ChartArea.ChartCategoryAxes</a>
<a href="#">ChartArea.ChartElementPosition</a>
<a href="#">ChartArea.ChartInnerPlotPosition</a>
<a href="#">ChartArea.ChartThreeDProperties</a>
<a href="#">ChartArea.ChartValueAxes</a>
<a href="#">ChartArea.EquallySizedAxesFont</a>
<a href="#">ChartArea.Hidden</a>
<a href="#">ChartArea.Style</a>

The following is the XML Schema definition of the **ChartArea** element in RDL 2008/01.

```
<xsd:complexType name="ChartAreaType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartCategoryAxes" type="ChartCategoryAxesType"
      minOccurs="0" />
    <xsd:element name="ChartValueAxes" type="ChartValueAxesType" minOccurs="0" />
    <xsd:element name="ChartThreeDProperties" type="ChartThreeDPropertiesType"
      minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="AlignOrientation" type="xsd:string" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

```

<xsd:element name="ChartAlignType" type="ChartAlignTypeType" minOccurs="0" />
<xsd:element name="ChartElementPosition" type="ChartElementPositionType"
minOccurs="0" />
<xsd:element name="ChartInnerPlotPosition" type="ChartElementPositionType"
minOccurs="0" />
<xsd:element name="AlignWithChartArea" type="xsd:string" minOccurs="0" />
<xsd:element name="EquallySizedAxesFont" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **ChartArea** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="ChartAreaType">
<xsd:choice minOccurs="0" maxOccurs="unbounded">
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="ChartCategoryAxes" type="ChartCategoryAxesType"
minOccurs="0" />
<xsd:element name="ChartValueAxes" type="ChartValueAxesType" minOccurs="0" />
<xsd:element name="ChartThreeDProperties" type="ChartThreeDPropertiesType"
minOccurs="0" />
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="AlignOrientation" type="xsd:string" minOccurs="0" />
<xsd:element name="ChartAlignType" type="ChartAlignTypeType" minOccurs="0" />
<xsd:element name="ChartElementPosition" type="ChartElementPositionType"
minOccurs="0" />
<xsd:element name="ChartInnerPlotPosition" type="ChartElementPositionType"
minOccurs="0" />
<xsd:element name="AlignWithChartArea" type="xsd:string" minOccurs="0" />
<xsd:element name="EquallySizedAxesFont" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.117.1 ChartArea.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartArea.Name** attribute specifies the name for a [ChartArea](#). This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique among the **ChartArea.Name** values in the parent collection.

Following is the parent element of the **ChartArea.Name** attribute.

Parent elements
ChartArea

The following is the XML Schema definition of the **ChartArea.Name** attribute.

```

<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />

```

## 2.117.2 ChartArea.AlignOrientation

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartArea.AlignOrientation** element specifies in which directions a [ChartArea](#) is aligned with the target **chart area**. This element is optional. If the **ChartArea.AlignOrientation** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**None**: Specifies that no alignment is applied.

**Vertical**: Specifies that vertical alignment is applied.

**Horizontal**: Specifies that horizontal alignment is applied.

**All**: Specifies that both vertical and horizontal alignment is applied.

If the **ChartArea.AlignOrientation** element is not present, its value is interpreted as "None". If this element is present, [ChartArea.AlignWithChartArea](#) is to be specified.

Following is the parent element of the **ChartArea.AlignOrientation** element.

Parent elements
ChartArea

The following is the XML Schema definition of the **ChartArea.AlignOrientation** element.

```
<xsd:element name="AlignOrientation" type="xsd:string" minOccurs="0" />
```

## 2.117.3 ChartArea.AlignWithChartArea

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartArea.AlignWithChartArea** element specifies the name of a [ChartArea](#) with which to align this **chart area**. The **ChartArea.AlignWithChartArea** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). This element is expected to be present if [ChartArea.ChartAlignType](#) is specified.

Following is the parent element of the **ChartArea.AlignWithChartArea** element.

Parent elements
ChartArea

The following is the XML Schema definition of the **ChartArea.AlignWithChartArea** element.

```
<xsd:element name="AlignWithChartArea" type="xsd:string" minOccurs="0" />
```

## 2.117.4 ChartArea.ChartAlignType

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartArea.ChartAlignType** element specifies which aspects of a [ChartArea](#) are aligned with the target **chart area**. This element is optional. If this element is present, [ChartArea.AlignWithChartArea](#) is expected to be specified.

The **ChartArea.ChartAlignType** element is of type [ChartAlignType](#).

Following is the parent element of the **ChartArea.ChartAlignType** element.

Parent elements
<a href="#">ChartArea</a>

The following is the XML Schema definition of the **ChartArea.ChartAlignType** element.

```
<xsd:element name="ChartAlignType" type="ChartAlignTypeType" minOccurs="0" />
```

### 2.117.5 ChartArea.ChartCategoryAxes

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartArea.ChartCategoryAxes** element specifies the category axes for a [Chart](#). This element is optional. This element is of type [ChartCategoryAxes](#).

Following is the parent element of the **ChartArea.ChartCategoryAxes** element.

Parent elements
<a href="#">ChartArea</a>

The following is the XML Schema definition of the **ChartArea.ChartCategoryAxes** element.

```
<xsd:element name="ChartCategoryAxes" type="ChartCategoryAxesType" minOccurs="0" />
```

### 2.117.6 ChartArea.ChartElementPosition

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartArea.ChartElementPosition** element specifies a custom position for a [ChartArea](#). The **ChartArea.ChartElementPosition** element is optional. If this element is not present, automatic positioning is used. This element is of type [ChartElementPosition](#).

Following is the parent element of the **ChartArea.ChartElementPosition** element.

Parent elements
<a href="#">ChartArea</a>

The following is the XML Schema definition of the **ChartArea.ChartElementPosition** element.

```
<xsd:element name="ChartElementPosition" type="ChartElementPositionType" minOccurs="0" />
```

## 2.117.7 ChartArea.ChartInnerPlotPosition

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartArea.ChartInnerPlotPosition** element specifies a custom position for the inner **plot area**. The **ChartArea.ChartInnerPlotPosition** element is optional. If this element is not present, automatic positioning is used. This element is of type [ChartElementPosition](#).

Following is the parent element of the **ChartArea.ChartInnerPlotPosition** element.

Parent elements
<a href="#">ChartArea</a>

The following is the XML Schema definition of the **ChartArea.ChartInnerPlotPosition** element.

```
<xsd:element name="ChartInnerPlotPosition" type="ChartElementPositionType"
  minOccurs="0" />
```

## 2.117.8 ChartArea.ChartThreeDProperties

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartArea.ChartThreeDProperties** element specifies the three-dimensional layout properties of a [Chart](#). This element is optional. This element is of type [ChartThreeDProperties](#).

Following is the parent element of the **ChartArea.ChartThreeDProperties** element.

Parent elements
<a href="#">ChartArea</a>

The following is the XML Schema definition of the **ChartArea.ChartThreeDProperties** element.

```
<xsd:element name="ChartThreeDProperties" type="ChartThreeDPropertiesType"
  minOccurs="0" />
```

## 2.117.9 ChartArea.ChartValueAxes

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartArea.ChartValueAxes** element specifies the value axes for a [Chart](#). This element is optional. This element is of type [ChartValueAxes](#).

Following is the parent element of the **ChartArea.ChartValueAxes** element.

Parent elements
<a href="#">ChartArea</a>

The following is the XML Schema definition of the **ChartArea.ChartValueAxes** element.

```
<xsd:element name="ChartValueAxes" type="ChartValueAxesType" minOccurs="0" />
```

### 2.117.10 ChartArea.EquallySizedAxesFont

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartArea.EquallySizedAxesFont** element specifies whether the same font size is used for all axes. The **ChartArea.EquallySizedAxesFont** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

If there is at least one [ChartAxis.LabelsAutoFitDisabled](#) element specified or evaluated as true either in [ChartArea.ChartCategoryAxes](#) or [ChartArea.ChartValueAxes](#), the value of the **ChartArea.EquallySizedAxesFont** element is interpreted as false.

Following is the parent element of the **ChartArea.EquallySizedAxesFont** element.

Parent elements
<a href="#">ChartArea</a>

The following is the XML Schema definition of the **ChartArea.EquallySizedAxesFont** element.

```
<xsd:element name="EquallySizedAxesFont" type="xsd:string" minOccurs="0" />
```

### 2.117.11 ChartArea.Hidden

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartArea.Hidden** element specifies whether a [ChartArea](#) is hidden. The **ChartArea.Hidden** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartArea.Hidden** element.

Parent elements
<a href="#">ChartArea</a>

The following is the XML Schema definition of the **ChartArea.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

### 2.117.12 ChartArea.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartArea.Style** element specifies the style properties for a [ChartArea](#). This element is optional. This element is of type [Style](#).

Following is the parent element of the **ChartArea.Style** element.

Parent elements
<a href="#">ChartArea</a>

The following is the XML Schema definition of the **ChartArea.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.118 ChartAlignType

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAlignType** element specifies whether the vertical or horizontal aspects of a [ChartArea](#) are aligned with the target **chart area**.

The following are the parent and child elements of the **ChartAlignType** element.

Parent elements
<a href="#">ChartArea</a>

Child elements
<a href="#">ChartAlignType.AxesView</a>
<a href="#">ChartAlignType.Cursor</a>
<a href="#">ChartAlignType.InnerPlotPosition</a>
<a href="#">ChartAlignType.Position</a>

The following is the XML Schema definition of the **ChartAlignType** element in RDL 2008/01.

```
<xsd:complexType name="ChartAlignTypeType">  
  <xsd:choice minOccurs="0" maxOccurs="unbounded">  
    <xsd:element name="AxesView" type="xsd:string" minOccurs="0" />  
    <xsd:element name="Cursor" type="xsd:string" minOccurs="0" />  
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />  
    <xsd:element name="InnerPlotPosition" type="xsd:string" minOccurs="0" />  
    <xsd:any namespace="##other" processContents="skip" />  
  </xsd:choice>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartAlignType** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartAlignTypeType">  
  <xsd:choice minOccurs="0" maxOccurs="unbounded">  
    <xsd:element name="AxesView" type="xsd:string" minOccurs="0" />  
    <xsd:element name="Cursor" type="xsd:string" minOccurs="0" />  
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />  
    <xsd:element name="InnerPlotPosition" type="xsd:string" minOccurs="0" />  
    <xsd:any namespace="##other" processContents="lax" />  
  </xsd:choice>  
  <xsd:anyAttribute namespace="##other" processContents="lax" />  
</xsd:complexType>
```

### 2.118.1 ChartAlignType.AxesView

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAlignType.AxesView** element specifies whether chart areas align on axes views. The **ChartAlignType.AxesView** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAlignType.AxesView** element.

Parent elements
<a href="#">ChartAlignType</a>

The following is the XML Schema definition of the **ChartAlignType.AxesView** element.

```
<xsd:element name="AxesView" type="xsd:string" minOccurs="0" />
```

### 2.118.2 ChartAlignType.Cursor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAlignType.Cursor** element specifies whether chart areas align on cursors. The **ChartAlignType.Cursor** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAlignType.Cursor** element.

Parent elements
<a href="#">ChartAlignType</a>

The following is the XML Schema definition of the **ChartAlignType.Cursor** element.

```
<xsd:element name="Cursor" type="xsd:string" minOccurs="0" />
```

### 2.118.3 ChartAlignType.InnerPlotPosition

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAlignType.InnerPlotPosition** element specifies whether chart areas align on inner plot positions. The **ChartAlignType.InnerPlotPosition** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAlignType.InnerPlotPosition** element.

Parent elements
<a href="#">ChartAlignType</a>

The following is the XML Schema definition of the **ChartAlignType.InnerPlotPosition** element.

```
<xsd:element name="InnerPlotPosition" type="xsd:string" minOccurs="0" />
```



## 2.118.4 ChartAlignType.Position

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAlignType.Position** element specifies whether chart areas align on chart area positions. The **ChartAlignType.Position** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAlignType.Position** element.

Parent elements
<a href="#">ChartAlignType</a>

The following is the XML Schema definition of the **ChartAlignType.Position** element.

```
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
```

## 2.119 ChartCategoryAxes

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartCategoryAxes** element specifies the set of [ChartAxis](#) instances as category axes (x-axes) for a [ChartArea](#).

The **ChartCategoryAxes** element SHOULD NOT contain more than one **ChartAxis** that has [ChartAxis.Location](#) set to "Default". Additionally, the **ChartCategoryAxes** element SHOULD NOT contain more than one **ChartAxis** that has **ChartAxis.Location** set to "Opposite".

The following are the parent and child elements of the **ChartCategoryAxes** element.

Parent elements
<a href="#">ChartArea</a>

Child elements
<a href="#">ChartCategoryAxes.ChartAxis</a>

The following is the XML Schema definition of the **ChartCategoryAxes** element in RDL 2008/01.

```
<xsd:complexType name="ChartCategoryAxesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartAxis" type="ChartAxisType" maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartCategoryAxes** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartCategoryAxesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
```

```

    <xsd:element name="ChartAxis" type="ChartAxisType" maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.119.1 ChartCategoryAxes.ChartAxis

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartCategoryAxes.ChartAxis** element specifies a [ChartAxis](#) for a [ChartArea](#). This element is optional and is of type **ChartAxis**.

Following is the parent element of the **ChartCategoryAxes.ChartAxis** element.

Parent elements
<a href="#">ChartCategoryAxes</a>

The following is the XML Schema definition of the **ChartCategoryAxes.ChartAxis** element.

```

<xsd:element name="ChartAxis" type="ChartAxisType" maxOccurs="unbounded" />

```

### 2.120 ChartAxis

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis** element specifies properties for labels, titles, and gridlines along an axis of a [Chart](#).

Following are the parent elements, attributes, and child elements of the **ChartAxis** element.

Parent elements
<a href="#">ChartCategoryAxes</a>
<a href="#">ChartValueAxes</a>

Attributes
<a href="#">ChartAxis.Name</a>

Child elements
<a href="#">ChartAxis.AllowLabelRotation</a>
<a href="#">ChartAxis.Angle</a>
<a href="#">ChartAxis.Arrows</a>
<a href="#">ChartAxis.ChartAxisScaleBreak</a>
<a href="#">ChartAxis.ChartAxisTitle</a>

<b>Child elements</b>
<a href="#">ChartAxis.ChartMajorGridLines</a>
<a href="#">ChartAxis.ChartMajorTickMarks</a>
<a href="#">ChartAxis.ChartMinorGridLines</a>
<a href="#">ChartAxis.ChartMinorTickMarks</a>
<a href="#">ChartAxis.ChartStripLines</a>
<a href="#">ChartAxis.CrossAt</a>
<a href="#">ChartAxis.CustomProperties</a>
<a href="#">ChartAxis.HideEndLabels</a>
<a href="#">ChartAxis.HideLabels</a>
<a href="#">ChartAxis.IncludeZero</a>
<a href="#">ChartAxis.Interlaced</a>
<a href="#">ChartAxis.InterlacedColor</a>
<a href="#">ChartAxis.Interval</a>
<a href="#">ChartAxis.IntervalOffset</a>
<a href="#">ChartAxis.IntervalOffsetType</a>
<a href="#">ChartAxis.IntervalType</a>
<a href="#">ChartAxis.LabelInterval</a>
<a href="#">ChartAxis.LabelIntervalOffset</a>
<a href="#">ChartAxis.LabelIntervalOffsetType</a>
<a href="#">ChartAxis.LabelIntervalType</a>
<a href="#">ChartAxis.LabelsAutoFitDisabled</a>
<a href="#">ChartAxis.Location</a>
<a href="#">ChartAxis.LogBase</a>
<a href="#">ChartAxis.LogScale</a>
<a href="#">ChartAxis.Margin</a>
<a href="#">ChartAxis.MarksAlwaysAtPlotEdge</a>
<a href="#">ChartAxis.MaxFontSize</a>
<a href="#">ChartAxis.Maximum</a>
<a href="#">ChartAxis.MinFontSize</a>
<a href="#">ChartAxis.Minimum</a>
<a href="#">ChartAxis.OffsetLabels</a>
<a href="#">ChartAxis.PreventFontGrow</a>

Child elements
<a href="#">ChartAxis.PreventFontShrink</a>
<a href="#">ChartAxis.PreventLabelOffset</a>
<a href="#">ChartAxis.PreventWordWrap</a>
<a href="#">ChartAxis.Reverse</a>
<a href="#">ChartAxis.Scalar</a>
<a href="#">ChartAxis.Style</a>
<a href="#">ChartAxis.VariableAutoInterval</a>
<a href="#">ChartAxis.Visible</a>

The following is the XML Schema definition of the **ChartAxis** element in RDL 2008/01.

```

<xsd:complexType name="ChartAxisType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Visible" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartAxisTitle" type="ChartAxisTitleType" minOccurs="0" />
    <xsd:element name="Margin" type="xsd:string" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:element name="VariableAutoInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelIntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelIntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelIntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartMajorGridLines" type="ChartGridLinesType" minOccurs="0" />
    <xsd:element name="ChartMinorGridLines" type="ChartGridLinesType" minOccurs="0" />
    <xsd:element name="ChartMajorTickMarks" type="ChartTickMarksType" minOccurs="0" />
    <xsd:element name="ChartMinorTickMarks" type="ChartTickMarksType" minOccurs="0" />
    <xsd:element name="MarksAlwaysAtPlotEdge" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reverse" type="xsd:string" minOccurs="0" />
    <xsd:element name="CrossAt" type="xsd:string" minOccurs="0" />
    <xsd:element name="Location" type="xsd:string" minOccurs="0" />
    <xsd:element name="Interlaced" type="xsd:string" minOccurs="0" />
    <xsd:element name="InterlacedColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartStripLines" type="ChartStripLinesType" minOccurs="0" />
    <xsd:element name="Arrows" type="xsd:string" minOccurs="0" />
    <xsd:element name="Scalar" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Minimum" type="xsd:string" minOccurs="0" />
    <xsd:element name="Maximum" type="xsd:string" minOccurs="0" />
    <xsd:element name="LogScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="LogBase" type="xsd:string" minOccurs="0" />
    <xsd:element name="HideLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
    <xsd:element name="PreventFontShrink" type="xsd:string" minOccurs="0" />
    <xsd:element name="PreventFontGrow" type="xsd:string" minOccurs="0" />
    <xsd:element name="PreventLabelOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="PreventWordWrap" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowLabelRotation" type="xsd:string" minOccurs="0" />
    <xsd:element name="IncludeZero" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelsAutoFitDisabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaxFontSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="HideEndLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartAxisScaleBreak" type="ChartAxisScaleBreakType" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>

```

```

    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **ChartAxis** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="ChartAxisType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Visible" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartAxisTitle" type="ChartAxisTitleType" minOccurs="0" />
    <xsd:element name="Margin" type="xsd:string" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:element name="VariableAutoInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelIntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelIntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelIntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartMajorGridLines" type="ChartGridLinesType" minOccurs="0" />
    <xsd:element name="ChartMinorGridLines" type="ChartGridLinesType" minOccurs="0" />
    <xsd:element name="ChartMajorTickMarks" type="ChartTickMarksType" minOccurs="0" />
    <xsd:element name="ChartMinorTickMarks" type="ChartTickMarksType" minOccurs="0" />
    <xsd:element name="MarksAlwaysAtPlotEdge" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reverse" type="xsd:string" minOccurs="0" />
    <xsd:element name="CrossAt" type="xsd:string" minOccurs="0" />
    <xsd:element name="Location" type="xsd:string" minOccurs="0" />
    <xsd:element name="Interlaced" type="xsd:string" minOccurs="0" />
    <xsd:element name="InterlacedColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartStripLines" type="ChartStripLinesType" minOccurs="0" />
    <xsd:element name="Arrows" type="xsd:string" minOccurs="0" />
    <xsd:element name="Scalar" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Minimum" type="xsd:string" minOccurs="0" />
    <xsd:element name="Maximum" type="xsd:string" minOccurs="0" />
    <xsd:element name="LogScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="LogBase" type="xsd:string" minOccurs="0" />
    <xsd:element name="HideLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
    <xsd:element name="PreventFontShrink" type="xsd:string" minOccurs="0" />
    <xsd:element name="PreventFontGrow" type="xsd:string" minOccurs="0" />
    <xsd:element name="PreventLabelOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="PreventWordWrap" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowLabelRotation" type="xsd:string" minOccurs="0" />
    <xsd:element name="IncludeZero" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelsAutoFitDisabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaxFontSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="HideEndLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartAxisScaleBreak" type="ChartAxisScaleBreakType" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.120.1 ChartAxis.Name

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.Name** attribute specifies the name for a [ChartAxis](#). This attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique among the **ChartAxis.Name** values in the parent collection.

Following is the parent element of the **ChartAxis.Name** attribute.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.120.2 ChartAxis.AllowLabelRotation

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.AllowLabelRotation** element specifies the "step" by which axis labels can be incrementally rotated to fit within a [Chart](#). This element is optional. If the **ChartAxis.AllowLabelRotation** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Rotate90:** Specifies that axis labels can rotate in 90-degree increments.

**Rotate30:** Specifies that axis labels can rotate in 30-degree increments.

**Rotate45:** Specifies that axis labels can rotate in 45-degree increments.

**None:** Specifies that axis labels cannot rotate.

If the **ChartAxis.AllowLabelRotation** element is not present, its value is interpreted as "Rotate90".

Following is the parent element of the **ChartAxis.AllowLabelRotation** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.AllowLabelRotation** element.

```
<xsd:element name="AllowLabelRotation" type="xsd:string" minOccurs="0" />
```

### 2.120.3 ChartAxis.Angle

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.Angle** element specifies the angle at which to display axis labels in a [Chart](#). The value of the **ChartAxis.Angle** element MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be equal to or greater than -90 and less than or equal to 90. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **ChartAxis.Angle** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.Angle** element.

```
<xsd:element name="Angle" type="xsd:string" minOccurs="0" />
```

#### 2.120.4 ChartAxis.Arrows

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.Arrows** element specifies the type of arrows for axis labels in a [Chart](#). The **ChartAxis.Arrows** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**None:** Specifies no arrows.

**Triangle:** Specifies triangle arrows.

**SharpTriangle:** Specifies sharp triangle arrows.

**Lines:** Specifies lines only.

If the **ChartAxis.Arrows** element is not present, its value is interpreted as "None".

Following is the parent element of the **ChartAxis.Arrows** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.Arrows** element.

```
<xsd:element name="Arrows" type="xsd:string" minOccurs="0" />
```

#### 2.120.5 ChartAxis.ChartAxisScaleBreak

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.ChartAxisScaleBreak** element specifies the **scale break** behavior of a [ChartAxis](#). This element is optional. The element is of type [ChartAxisScaleBreak](#).

Following is the parent element of the **ChartAxis.ChartAxisScaleBreak** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.ChartAxisScaleBreak** element.

```
<xsd:element name="ChartAxisScaleBreak" type="ChartAxisScaleBreakType"
  minOccurs="0" />
```

## 2.120.6 ChartAxis.ChartAxisTitle

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.ChartAxisTitle** element specifies the title of a [ChartAxis](#). This element is optional. This element is of type [ChartAxisTitle](#).

Following is the parent element of the **ChartAxis.ChartAxisTitle** element.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.ChartAxisTitle** element.

```
<xsd:element name="ChartAxisTitle" type="ChartAxisTitleType" minOccurs="0" />
```

## 2.120.7 ChartAxis.ChartMajorGridLines

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.ChartMajorGridLines** element specifies the major gridlines for a [ChartAxis](#). This element is optional. This element is of type [ChartGridLines](#).

Following is the parent element of the **ChartAxis.ChartMajorGridLines** element.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.ChartMajorGridLines** element.

```
<xsd:element name="ChartMajorGridLines" type="ChartGridLinesType" minOccurs="0" />
```

## 2.120.8 ChartAxis.ChartMajorTickMarks

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.ChartMajorTickMarks** element specifies the major tick marks for a [ChartAxis](#). This element is optional. This element is of type [ChartTickMarks](#).

Following is the parent element of the **ChartAxis.ChartMajorTickMarks** element.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.ChartMajorTickMarks** element.

```
<xsd:element name="ChartMajorTickMarks" type="ChartTickMarksType" minOccurs="0" />
```



## 2.120.9 ChartAxis.ChartMinorGridLines

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.ChartMinorGridLines** element specifies the minor gridlines for a [ChartAxis](#). This element is optional. This element is of type [ChartGridLines](#).

Following is the parent element of the **ChartAxis.ChartMinorGridLines** element.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.ChartMinorGridLines** element.

```
<xsd:element name="ChartMinorGridLines" type="ChartGridLinesType" minOccurs="0" />
```

## 2.120.10 ChartAxis.ChartMinorTickMarks

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.ChartMinorTickMarks** element specifies the minor tick marks for a [ChartAxis](#). This element is optional. This element is of type [ChartTickMarks](#).

Following is the parent element of the **ChartAxis.ChartMinorTickMarks** element.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.ChartMinorTickMarks** element.

```
<xsd:element name="ChartMinorTickMarks" type="ChartTickMarksType" minOccurs="0" />
```

## 2.120.11 ChartAxis.ChartStripLines

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.ChartStripLines** element specifies custom **strip lines** for a [ChartAxis](#). The **ChartAxis.ChartStripLines** element is optional. This element is of type [ChartStripLines](#).

Following is the parent element of the **ChartAxis.ChartStripLines** element.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.ChartStripLines** element.

```
<xsd:element name="ChartStripLines" type="ChartStripLinesType" minOccurs="0" />
```

## 2.120.12 ChartAxis.CrossAt

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.CrossAt** element specifies the value at which an axis crosses the other axis. This element is optional. If the **ChartAxis.CrossAt** element is present, its value MUST be a **Numeric** or [DateTime](#) ([XMLSCHEMA2] section 3.2.7) constant or an expression that evaluates to a **Variant** of type **Numeric** or **DateTime**. If this element is not present, the behavior is the default behavior of the chart type.

Following is the parent element of the **ChartAxis.CrossAt** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.CrossAt** element.

```
<xsd:element name="CrossAt" type="xsd:string" minOccurs="0" />
```

### 2.120.13 ChartAxis.CustomProperties

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.CustomProperties** element specifies the custom properties for a [ChartAxis](#). This element is optional. This element is of type [CustomProperties](#).

Following is the parent element of the **ChartAxis.CustomProperties** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

### 2.120.14 ChartAxis.HideEndLabels

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.HideEndLabels** element specifies whether labels are hidden at axis ends. The **ChartAxis.HideEndLabels** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAxis.HideEndLabels** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.HideEndLabels** element.

```
<xsd:element name="HideEndLabels" type="xsd:string" minOccurs="0" />
```

## 2.120.15 ChartAxis.HideLabels

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.HideLabels** element specifies whether axis labels are hidden. The **ChartAxis.HideLabels** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAxis.HideLabels** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.HideLabels** element.

```
<xsd:element name="HideLabels" type="xsd:string" minOccurs="0" />
```

## 2.120.16 ChartAxis.IncludeZero

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.IncludeZero** element specifies whether a [ChartAxis](#) always includes a value of zero. The **ChartAxis.IncludeZero** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**.

If this element is not present, its value is interpreted as true.

Following is the parent element of the **ChartAxis.IncludeZero** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.IncludeZero** element.

```
<xsd:element name="IncludeZero" type="xsd:string" minOccurs="0" />
```

## 2.120.17 ChartAxis.Interlaced

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.Interlaced** element specifies whether **strip lines** are drawn at an interval of every other gridline for a [ChartAxis](#). If gridlines are not used for the axis, the axis tick marks or labels are used to determine the interval of the interlaced strip lines.

The **ChartAxis.Interlaced** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAxis.Interlaced** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.Interlaced** element.

```
<xsd:element name="Interlaced" type="xsd:string" minOccurs="0" />
```

### 2.120.18 ChartAxis.InterlacedColor

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.InterlacedColor** element specifies the color of interlaced strips for a [ChartAxis](#). The **ChartAxis.InterlacedColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. If this element is not present, its value is interpreted as an empty color.

Following is the parent element of the **ChartAxis.InterlacedColor** element.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.InterlacedColor** element.

```
<xsd:element name="InterlacedColor" type="xsd:string" minOccurs="0" />
```

### 2.120.19 ChartAxis.Interval

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.Interval** element specifies the interval between gridlines, tick marks, and labels of a [ChartAxis](#). This element is optional.

If the **ChartAxis.Interval** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If the value of this element is specified as 0, its value is determined by the axis. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **ChartAxis.Interval** element.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.Interval** element.

```
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
```

### 2.120.20 ChartAxis.IntervalOffset

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.IntervalOffset** element specifies the offset for the first tick mark from the axis minimum. This element is optional. If the **ChartAxis.IntervalOffset** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **ChartAxis.IntervalOffset** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartAxis</a>
---------------------------

The following is the XML Schema definition of the **ChartAxis.IntervalOffset** element.

```
<xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
```

### 2.120.21 **ChartAxis.IntervalOffsetType**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.IntervalOffsetType** element specifies the unit to use for the [ChartAxis.IntervalOffset](#) element. The **ChartAxis.IntervalOffsetType** element is optional. If the **ChartAxis.IntervalOffsetType** element is present, its value MUST be a [String \(XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Auto:** Specifies that the interval offset unit is determined automatically based on the data that is plotted against the axis.

**Number:** Specifies that the interval offset is numeric.

**Years:** Specifies that the interval offset is years.

**Months:** Specifies that the interval offset is months.

**Weeks:** Specifies that the interval offset is weeks.

**Days:** Specifies that the interval offset is days.

**Hours:** Specifies that the interval offset is hours.

**Minutes:** Specifies that the interval offset is minutes.

**Seconds:** Specifies that the interval offset is seconds.

**Milliseconds:** Specifies that the interval offset is milliseconds.

If the **ChartAxis.IntervalOffsetType** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartAxis.IntervalOffsetType** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartAxis</a>
---------------------------

The following is the XML Schema definition of the **ChartAxis.IntervalOffsetType** element.

```
<xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
```

### 2.120.22 **ChartAxis.IntervalType**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.IntervalType** element specifies the unit to use for the [ChartAxis.Interval](#) element. The **ChartAxis.IntervalType** element is optional. If the **ChartAxis.IntervalType** element is

present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Auto:** Specifies that the interval unit is determined based on the data plotted against the axis.

**Number:** Specifies that the interval is numeric.

**Years:** Specifies that the interval is years.

**Months:** Specifies that the interval is months.

**Weeks:** Specifies that the interval is weeks.

**Days:** Specifies that the interval is days.

**Hours:** Specifies that the interval is hours.

**Minutes:** Specifies that the interval is minutes.

**Seconds:** Specifies that the interval is seconds.

**Milliseconds:** Specifies that the interval is milliseconds.

If the **ChartAxis.IntervalType** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartAxis.IntervalType** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.IntervalType** element.

```
<xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
```

### 2.120.23 ChartAxis.LabelInterval

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.LabelInterval** element specifies the interval between chart axis labels. The **ChartAxis.LabelInterval** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If the value of this element is specified as 0 or if this element is not present, the value of the [ChartAxis.Interval](#) element MUST be used.

Following is the parent element of the **ChartAxis.LabelInterval** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.LabelInterval** element.

```
<xsd:element name="LabelInterval" type="xsd:string" minOccurs="0" />
```

## 2.120.24 ChartAxis.LabelIntervalOffset

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.LabelIntervalOffset** element specifies the offset for the first axis label from the axis minimum. The **ChartAxis.LabelIntervalOffset** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If the value of this element is specified as 0 or if this element is not present, the value of the [ChartAxis.IntervalOffset](#) element MUST be used.

Following is the parent element of the **ChartAxis.LabelIntervalOffset** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.LabelIntervalOffset** element.

```
<xsd:element name="LabelIntervalOffset" type="xsd:string" minOccurs="0" />
```

## 2.120.25 ChartAxis.LabelIntervalOffsetType

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.LabelIntervalOffsetType** element specifies the unit to use for the [ChartAxis.LabelIntervalOffset](#) element. This element is optional. If the **ChartAxis.LabelIntervalOffsetType** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies that the value of the [ChartAxis.IntervalOffsetType](#) element is used.

**Auto:** Specifies that the label interval offset unit is determined based on the data plotted against the axis.

**Number:** Specifies that the label interval offset is numeric.

**Years:** Specifies that the label interval offset is years.

**Months:** Specifies that the label interval offset is months.

**Weeks:** Specifies that the label interval offset is weeks.

**Days:** Specifies that the label interval offset is days.

**Hours:** Specifies that the label interval offset is hours.

**Minutes:** Specifies that the label interval offset is minutes.

**Seconds:** Specifies that the label interval offset is seconds.

**Milliseconds:** Specifies that the label interval offset is milliseconds.

If the **ChartAxis.LabelIntervalOffsetType** element is not present, its value is interpreted as "Default".

Following is the parent element of the **ChartAxis.IntervalOffsetType** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartAxis</a>
---------------------------

The following is the XML Schema definition of the **ChartAxis.LabelIntervalOffsetType** element.

```
<xsd:element name="LabelIntervalOffsetType" type="xsd:string" minOccurs="0" />
```

## 2.120.26 ChartAxis.LabelIntervalType

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.LabelIntervalType** element specifies the unit to use for the [ChartAxis.LabelInterval](#) element. This element is optional. If the **ChartAxis.LabelIntervalType** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies that the value of the [ChartAxis.IntervalType](#) element is used.

**Auto:** Specifies that the label interval unit is determined based on the data plotted against the axis.

**Number:** Specifies that the label interval is numeric.

**Years:** Specifies that the label interval is years.

**Months:** Specifies that the label interval is months.

**Weeks:** Specifies that the label interval is weeks.

**Days:** Specifies that the label interval is days.

**Hours:** Specifies that the label interval is hours.

**Minutes:** Specifies that the label interval is minutes.

**Seconds:** Specifies that the label interval is seconds.

**Milliseconds:** Specifies that the label interval is milliseconds.

If the **ChartAxis.LabelIntervalType** element is not present, its value is interpreted as "Default".

Following is the parent element of the **ChartAxis.LabelIntervalType** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartAxis</a>
---------------------------

The following is the XML Schema definition of the **ChartAxis.LabelIntervalType** element.

```
<xsd:element name="LabelIntervalType" type="xsd:string" minOccurs="0" />
```

## 2.120.27 ChartAxis.LabelsAutoFitDisabled

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)



The **ChartAxis.LabelsAutoFitDisabled** element specifies whether axis labels are not automatically adjusted to fit. This element is optional. If the **ChartAxis.LabelsAutoFitDisabled** element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean** value. If the value of the **ChartAxis.LabelsAutoFitDisabled** element is false, labels MUST adjust to fit. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAxis.LabelsAutoFitDisabled** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.LabelsAutoFitDisabled** element.

```
<xsd:element name="LabelsAutoFitDisabled" type="xsd:string" minOccurs="0" />
```

### 2.120.28 ChartAxis.Location

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.Location** element specifies whether a [ChartAxis](#) is drawn on the default side or on the opposite side. This element is optional. If the **ChartAxis.Location** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String** value. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default** (Default): Specifies that the axis is drawn on the default side.

**Opposite**: Specifies that the axis is drawn on the opposite side.

If the **ChartAxis.Location** element is not present, its value is interpreted as "Default".

Following is the parent element of the **ChartAxis.Location** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.Location** element.

```
<xsd:element name="Location" type="xsd:string" minOccurs="0" />
```

### 2.120.29 ChartAxis.LogBase

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.LogBase** element specifies the base to use for a logarithmic scale. The **ChartAxis.LogBase** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 10. The value of this element SHOULD be greater than 2.

Following is the parent element of the **ChartAxis.LogBase** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.LogBase** element.

```
<xsd:element name="LogBase" type="xsd:string" minOccurs="0" />
```

### 2.120.30 ChartAxis.LogScale

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.LogScale** element specifies whether an axis is logarithmic. The **ChartAxis.LogScale** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAxis.LogScale** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.LogScale** element.

```
<xsd:element name="LogScale" type="xsd:string" minOccurs="0" />
```

### 2.120.31 ChartAxis.Margin

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.Margin** element specifies whether an axis margin will be created. The size of the margin MUST be automatically generated based on the axis scale and the number of **data points**. This element is optional.

If the **ChartAxis.Margin** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Auto:** Specifies that the margins are included based on the [ChartSeries.Type](#) or [ChartSeries.Subtype](#).

**True:** Specifies that the axis has a margin.

**False:** Specifies that the axis has no margin.

If the **ChartAxis.Margin** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartAxis.Margin** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.Margin** element.

```
<xsd:element name="Margin" type="xsd:string" minOccurs="0" />
```

### 2.120.32 ChartAxis.MarksAlwaysAtPlotEdge

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.MarksAlwaysAtPlotEdge** element specifies whether marks stay with the edge of a **plot area** rather than moving with the [ChartAxis](#). The **ChartAxis.MarksAlwaysAtPlotEdge** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAxis.MarksAlwaysAtPlotEdge** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.MarksAlwaysAtPlotEdge** element.

```
<xsd:element name="MarksAlwaysAtPlotEdge" type="xsd:string" minOccurs="0" />
```

### 2.120.33 ChartAxis.MaxFontSize

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.MaxFontSize** element specifies the maximum font size when auto-fitting axis labels. The **ChartAxis.MaxFontSize** element is optional. If this element is present, its value MUST be an [RdSize](#) or an expression that evaluates to an **RdSize**. If this element is not present, its value is interpreted as 10pt.

Following is the parent element of the **ChartAxis.MaxFontSize** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.MaxFontSize** element.

```
<xsd:element name="MaxFontSize" type="xsd:string" minOccurs="0" />
```

### 2.120.34 ChartAxis.Maximum

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.Maximum** element specifies the maximum value for a [ChartAxis](#). This element is optional. If the **ChartAxis.Maximum** element is present, its value MUST be a **Numeric** or [DateTime](#) ([\[XMLSCHEMA2\]](#) section 3.2.7) constant or an expression that evaluates to a **Variant** of type **Numeric** or **DateTime**. If this element is not specified, the axis MUST calculate its scale automatically.

Following is the parent element of the **ChartAxis.Maximum** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.Maximum** element.

```
<xsd:element name="Maximum" type="xsd:string" minOccurs="0" />
```

### 2.120.35 ChartAxis.MinFontSize

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.MinFontSize** element specifies the minimum font size when auto-fitting axis labels. The **ChartAxis.MinFontSize** element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**. If this element is not present, its value is interpreted as 6pt.

Following is the parent element of the **ChartAxis.MinFontSize** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.MinFontSize** element.

```
<xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
```

### 2.120.36 ChartAxis.Minimum

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.Minimum** element specifies the minimum value for a [ChartAxis](#). The **ChartAxis.Minimum** element is optional. If this element is present, its value MUST be a **Numeric** or [DateTime](#) ([\[XMLSCHEMA2\]](#) section 3.2.7) constant or an expression that evaluates to a **Variant** of type **Numeric** or **DateTime**. If this element is not specified, the axis MUST calculate its scale automatically.

Following is the parent element of the **ChartAxis.Minimum** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.Minimum** element.

```
<xsd:element name="Minimum" type="xsd:string" minOccurs="0" />
```

### 2.120.37 ChartAxis.OffsetLabels

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.OffsetLabels** element specifies whether axis labels are offset. The **ChartAxis.OffsetLabels** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAxis.OffsetLabels** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartAxis</a>
---------------------------

The following is the XML Schema definition of the **ChartAxis.OffsetLabels** element.

```
<xsd:element name="OffsetLabels" type="xsd:string" minOccurs="0" />
```

### 2.120.38 ChartAxis.PreventFontGrow

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.PreventFontGrow** element specifies whether the font size for an axis label is not increased to fit within a [Chart](#). The **ChartAxis.PreventFontGrow** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAxis.PreventFontGrow** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartAxis</a>
---------------------------

The following is the XML Schema definition of the **ChartAxis.PreventFontGrow** element.

```
<xsd:element name="PreventFontGrow" type="xsd:string" minOccurs="0" />
```

### 2.120.39 ChartAxis.PreventFontShrink

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.PreventFontShrink** element specifies whether the font size for an axis label is not reduced to fit within a [Chart](#). The **ChartAxis.PreventFontShrink** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of **ChartAxis.PreventFontShrink** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartAxis</a>
---------------------------

The following is the XML Schema definition of the **ChartAxis.PreventFontShrink** element.

```
<xsd:element name="PreventFontShrink" type="xsd:string" minOccurs="0" />
```

### 2.120.40 ChartAxis.PreventLabelOffset

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.PreventLabelOffset** element specifies whether axis labels are staggered to fit within a [Chart](#). The **ChartAxis.PreventLabelOffset** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAxis.PreventLabelOffset** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.PreventLabelOffset** element.

```
<xsd:element name="PreventLabelOffset" type="xsd:string" minOccurs="0" />
```

### 2.120.41 ChartAxis.PreventWordWrap

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.PreventWordWrap** element specifies whether axis labels are not word-wrapped to fit within a [Chart](#). The **ChartAxis.PreventWordWrap** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAxis.PreventWordWrap** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.PreventWordWrap** element.

```
<xsd:element name="PreventWordWrap" type="xsd:string" minOccurs="0" />
```

### 2.120.42 ChartAxis.Reverse

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.Reverse** element specifies whether an axis is plotted in the reverse direction. The **ChartAxis.Reverse** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAxis.Reverse** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.Reverse** element.

```
<xsd:element name="Reverse" type="xsd:string" minOccurs="0" />
```

### 2.120.43 ChartAxis.Scalar

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.Scalar** element specifies whether the values along a chart axis are scalar values (that is, numeric or date) that are displayed on the [Chart](#) in a continuous axis. The type of scalar (**Date**,

[Integer](#) ([XMLSCHEMA2/2] section 3.3.17), or [Float](#) ([XMLSCHEMA2] section 3.2.4)) value is derived from the first non-**null value** found. All values are converted to that type. If any non-scalar value is present, the axis **MUST** revert to non-scalar.

The **ChartAxis.Scalar** element is optional. If this element is present, its value **MUST** be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false.

The **ChartAxis.Scalar** element is interpreted as true if both of the following statements are true:

- This is a [ChartAxis](#) in the [ChartCategoryAxes](#).
- Any [ChartSeries](#) plotted against this axis contains a [ChartDataPoint](#) with [ChartDataPointValues.X](#) defined.

Following is the parent element of the **ChartAxis.Scalar** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.Scalar** element.

```
<xsd:element name="Scalar" type="xsd:boolean" minOccurs="0" />
```

#### 2.120.44 **ChartAxis.Style**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.Style** element specifies the text style for the axis labels and axis line in a [Chart](#). This element is optional. This element is of type [Style](#).

Following is the parent element of the **ChartAxis.Style** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

#### 2.120.45 **ChartAxis.VariableAutoInterval**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxis.VariableAutoInterval** element specifies whether the value of [ChartAxis.Interval](#) is calculated automatically based on available size or if the value for **ChartAxis.Interval** is calculated based only on the data range.

The **ChartAxis.VariableAutoInterval** element is optional. If this element is present, its value **MUST** be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAxis.VariableAutoInterval** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.VariableAutoInterval** element.

```
<xsd:element name="VariableAutoInterval" type="xsd:string" minOccurs="0" />
```

## 2.120.46 ChartAxis.Visible

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxis.Visible** element specifies whether to display a [ChartAxis](#). The value of this element MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. This element is optional. If the **ChartAxis.Visible** element is present, its value MUST be one of the following:

**Auto:** Specifies that the axis is displayed if it is in use (such as if a series is plotted against it, or if it has the [ChartAxis.ChartAxisTitle](#) element specified).

**True:** Specifies that the axis is displayed.

**False:** Specifies that the axis is hidden.

If the **ChartAxis.Visible** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartAxis.Visible** element.

Parent elements
<a href="#">ChartAxis</a>

The following is the XML Schema definition of the **ChartAxis.Visible** element.

```
<xsd:element name="Visible" type="xsd:string" minOccurs="0" />
```

## 2.121 ChartAxisScaleBreak

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxisScaleBreak** element specifies **scale break** behavior and style for a [ChartAxis](#).

Following are the parent and child elements of the **ChartAxisScaleBreak** element.

Parent element
<a href="#">ChartAxis</a>

Child elements
<a href="#">ChartAxisScaleBreak.BreakLineStyle</a>
<a href="#">ChartAxisScaleBreak.CollapsibleSpaceThreshold</a>



Child elements
<a href="#">ChartAxisScaleBreak.Enabled</a>
<a href="#">ChartAxisScaleBreak.IncludeZero</a>
<a href="#">ChartAxisScaleBreak.MaxNumberOfBreaks</a>
<a href="#">ChartAxisScaleBreak.Spacing</a>
<a href="#">ChartAxisScaleBreak.Style</a>

The following is the XML Schema definition of the **ChartAxisScaleBreak** element in RDL 2008/01.

```
<xsd:complexType name="ChartAxisScaleBreakType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="BreakLineType" type="xsd:string" minOccurs="0" />
    <xsd:element name="CollapsibleSpaceThreshold" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaxNumberOfBreaks" type="xsd:string" minOccurs="0" />
    <xsd:element name="Spacing" type="xsd:string" minOccurs="0" />
    <xsd:element name="IncludeZero" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartAxisScaleBreak** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartAxisScaleBreakType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="BreakLineType" type="xsd:string" minOccurs="0" />
    <xsd:element name="CollapsibleSpaceThreshold" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaxNumberOfBreaks" type="xsd:string" minOccurs="0" />
    <xsd:element name="Spacing" type="xsd:string" minOccurs="0" />
    <xsd:element name="IncludeZero" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.121.1 ChartAxisScaleBreak.BreakLineType

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxisScaleBreak.BreakLineType** element specifies the type of line to show in the **scale break**. This element is optional. If the **ChartAxisScaleBreak.BreakLineType** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Ragged:** Specifies that the display is a ragged line.

**Straight:** Specifies that the display is a straight line.

**Wave:** Specifies that the display is a wavy line.

**None:** Specifies that a line for the scale break is not displayed.

If the **ChartAxisScaleBreak.BreakLineType** element is not present, its value is interpreted as "Ragged".

Following is the parent element of the **ChartAxisScaleBreak.BreakLineType** element.

Parent elements
<a href="#">ChartAxisScaleBreak</a>

The following is the XML Schema definition of the **ChartAxisScaleBreak.BreakLineType** element.

```
<xsd:element name="BreakLineType" type="xsd:string" minOccurs="0" />
```

### 2.121.2 ChartAxisScaleBreak.CollapsibleSpaceThreshold

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxisScaleBreak.CollapsibleSpaceThreshold** element specifies the percentage of empty space that is allowed on a [ChartAxis](#) before a **scale break** is triggered. This element is optional.

If the **ChartAxisScaleBreak.CollapsibleSpaceThreshold** element is present, its value **MUST** be an **Integer** or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 25.<23>

Following is the parent element of the **ChartAxisScaleBreak.CollapsibleSpaceThreshold** element.

Parent elements
<a href="#">ChartAxisScaleBreak</a>

The following is the XML Schema definition of the **ChartAxisScaleBreak.CollapsibleSpaceThreshold** element.

```
<xsd:element name="CollapsibleSpaceThreshold" type="xsd:string" minOccurs="0" />
```

### 2.121.3 ChartAxisScaleBreak.Enabled

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxisScaleBreak.Enabled** element specifies whether **scale breaks** are automatically applied. The **ChartAxisScaleBreak.Enabled** element is optional. If this element is present, its value **MUST** be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartAxisScaleBreak.Enabled** element.

Parent elements
<a href="#">ChartAxisScaleBreak</a>

The following is the XML Schema definition of the **ChartAxisScaleBreak.Enabled** element.

```
<xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
```

## 2.121.4 ChartAxisScaleBreak.IncludeZero

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxisScaleBreak.IncludeZero** element specifies whether a **scale break** is prohibited from spanning zero. This element is optional. If the **ChartAxisScaleBreak.IncludeZero** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Auto**: Specifies whether scale breaks are allowed to span zero based on the data plotted against the axis.

**True**: Specifies that a scale break is not allowed to span zero.

**False**: Specifies that a scale break is allowed to span zero.

If the **ChartAxisScaleBreak.IncludeZero** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartAxisScaleBreak.IncludeZero** element.

Parent elements
<a href="#">ChartAxisScaleBreak</a>

The following is the XML Schema definition of the **ChartAxisScaleBreak.IncludeZero** element.

```
<xsd:element name="IncludeZero" type="xsd:string" minOccurs="0" />
```

## 2.121.5 ChartAxisScaleBreak.MaxNumberOfBreaks

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxisScaleBreak.MaxNumberOfBreaks** element specifies the maximum number of **scale breaks** to apply.

The **ChartAxisScaleBreak.MaxNumberOfBreaks** element is optional. If this element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2](#) section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 2. The value of this element MUST be greater than or equal to 1. [<24>](#)

Following is the parent element of the **ChartAxisScaleBreak.MaxNumberOfBreaks** element.

Parent elements
<a href="#">ChartAxisScaleBreak</a>

The following is the XML Schema definition of the **ChartAxisScaleBreak.MaxNumberOfBreaks** element.

```
<xsd:element name="MaxNumberOfBreaks" type="xsd:string" minOccurs="0" />
```

## 2.121.6 ChartAxisScaleBreak.Spacing

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxisScaleBreak.Spacing** element specifies the amount of space to leave for a **scale break** as a percentage of the chart size. The **ChartAxisScaleBreak.Spacing** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 1.5. The value of this element MUST be greater than or equal to 0. <25>

Following is the parent element of the **ChartAxisScaleBreak.Spacing** element.

Parent elements
<a href="#">ChartAxisScaleBreak</a>

The following is the XML Schema definition of the **ChartAxisScaleBreak.Spacing** element.

```
<xsd:element name="Spacing" type="xsd:string" minOccurs="0" />
```

### 2.121.7 ChartAxisScaleBreak.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxisScaleBreak.Style** element specifies the style properties for a **scale break**. This element is optional. This element is of type [Style](#).

Following is the parent element of the **ChartAxisScaleBreak.Style** element.

Parent elements
<a href="#">ChartAxisScaleBreak</a>

The following is the XML Schema definition of the **ChartAxisScaleBreak.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.122 ChartAxisTitle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartAxisTitle** element specifies a title for a [ChartAxis](#).

Following are the parent and child elements of the **ChartAxisTitle** element.

Parent elements
ChartAxis

Child elements
<a href="#">ChartAxisTitle.Caption</a>
<a href="#">ChartAxisTitle.Position</a>
<a href="#">ChartAxisTitle.Style</a>
<a href="#">ChartAxisTitle.TextOrientation</a>

The following is the XML Schema definition of the **ChartAxisTitle** element in RDL 2008/01.

```
<xsd:complexType name="ChartAxisTitleType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Caption" type="StringLocIDType" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartAxisTitle** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartAxisTitleType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Caption" type="StringLocIDType" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.122.1 ChartAxisTitle.Caption

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxisTitle.Caption** element specifies the caption of an axis title. This element **MUST** be specified. The value of this element **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **ChartAxisTitle.Caption** element.

<b>Parent elements</b>
<a href="#">ChartAxisTitle</a>

The following is the XML Schema definition of the **ChartAxisTitle.Caption** element.

```
<xsd:element name="Caption" type="StringLocIDType" />
```

### 2.122.2 ChartAxisTitle.Position

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxisTitle.Position** element specifies the position of an axis title along the [ChartAxis](#). This element is optional. If the **ChartAxisTitle.Position** element is present, its value **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element **MUST** be one of the following or an expression that evaluates to one of the following:

**Center:** Specifies that the axis title is positioned on the center of the axis.

**Near:** Specifies that the axis title is positioned on the near side of the axis (the near side is the origin of the primary axes).

**Far:** Specifies that the axis title is positioned on the far side of the axis.

If the **ChartAxisTitle.Position** element is not present, its value is interpreted as "Center".

Following is the parent element of the **ChartAxisTitle.Position** element.

Parent elements
<a href="#">ChartAxisTitle</a>

The following is the XML Schema definition of the **ChartAxisTitle.Position** element.

```
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
```

### 2.122.3 ChartAxisTitle.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxisTitle.Style** element specifies the style properties for an axis title. This element is optional. This element is of type [Style](#).

Following is the parent element of the **ChartAxisTitle.Style** element.

Parent elements
<a href="#">ChartAxisTitle</a>

The following is the XML Schema definition of the **ChartAxisTitle.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.122.4 ChartAxisTitle.TextOrientation

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartAxisTitle.TextOrientation** element specifies the orientation of axis title text. This element is optional. If the **ChartAxisTitle.TextOrientation** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) value or an expression that evaluates to a **String** value. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Horizontal:** Specifies horizontal text.

**Rotated90:** Specifies vertical text, rotated 90 degrees.

**Rotated270:** Specifies vertical text, rotated 270 degrees.

**Stacked:** Specifies vertical text with no character rotation.

**Auto:** Specifies that the orientation is selected automatically based on context (for example, "Rotated270" is selected for titles that are docked on the left).

If the **ChartAxisTitle.TextOrientation** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartAxisTitle.TextOrientation** element.

Parent elements
<a href="#">ChartAxisTitle</a>

The following is the XML Schema definition of the **ChartAxisTitle.TextOrientation** element.

```
<xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
```

## 2.123 ChartGridLines

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartGridLines** element specifies gridlines along a [ChartAxis](#).

The following are the parent and child elements of the **ChartGridLines** element.

Parent elements
<a href="#">ChartAxis</a>

Child elements
<a href="#">ChartGridLines.Enabled</a>
<a href="#">ChartGridLines.Interval</a>
<a href="#">ChartGridLines.IntervalOffset</a>
<a href="#">ChartGridLines.IntervalOffsetType</a>
<a href="#">ChartGridLines.IntervalType</a>
<a href="#">ChartGridLines.Style</a>

The following is the XML Schema definition of the **ChartGridLines** element in RDL 2008/01.

```
<xsd:complexType name="ChartGridLinesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartGridLines** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartGridLinesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

```

<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
<xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
<xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
<xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.123.1 ChartGridLines.Enabled

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartGridLines.Enabled** element specifies whether axis gridlines are shown. This element is optional.

If the **ChartGridLines.Enabled** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**True:** Specifies that gridlines appear.

**False:** Specifies that gridlines do not appear.

**Auto:** Specifies that this value is interpreted as true for [ChartAxis.ChartMajorGridLines](#) and as false for [ChartAxis.ChartMinorGridLines](#).

If the **ChartGridLines.Enabled** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartGridLines.Enabled** element.

Parent elements
<a href="#">ChartGridLines</a>

The following is the XML Schema definition of the **ChartGridLines.Enabled** element.

```

<xsd:element name="Enabled" type="xsd:string" minOccurs="0" />

```

### 2.123.2 ChartGridLines.Interval

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartGridLines.Interval** element specifies the interval between axis gridlines. This element is optional.

If the **ChartGridLines.Interval** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is specified as 0, its value is determined by the axis interval specified in the [ChartAxis.Interval](#) element.

If the value of the **ChartGridLines.Interval** element evaluates to 0 or if this element is not present, the value of **ChartAxis.Interval** is used as the value of the **ChartGridLines.Interval** element.

Following is the parent element of the **ChartGridLines.Interval** element.



<b>Parent elements</b>
------------------------

<a href="#">ChartGridLines</a>
--------------------------------

The following is the XML Schema definition of the **ChartGridLines.Interval** element.

```
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
```

### 2.123.3 ChartGridLines.IntervalOffset

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartGridLines.IntervalOffset** element specifies the offset for the first gridline from the value of the **ChartAxis.Minimum** element. This element is optional.

If the **ChartGridLines.IntervalOffset** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\] section 3.2.4](#)) or an expression that evaluates to a **Float**. If the value of the **ChartGridLines.IntervalOffset** element evaluates to 0 or if this element is not present, the value of [ChartAxis.IntervalOffset](#) is used as the value for the **ChartGridLines.IntervalOffset** element.

Following is the parent element of the **ChartGridLines.IntervalOffset** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartGridLines</a>
--------------------------------

The following is the XML Schema definition of the **ChartGridLines.IntervalOffset** element.

```
<xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
```

### 2.123.4 ChartGridLines.IntervalOffsetType

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartGridLines.IntervalOffsetType** element specifies the unit to use for the [ChartGridLines.IntervalOffset](#) element. The **ChartGridLines.IntervalOffsetType** element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\] section 3.2.1](#)) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies that the value of [ChartAxis.IntervalOffsetType](#) is used.

**Auto:** Specifies that the interval offset unit is determined automatically based on the data plotted against the axis.

**Number:** Specifies that the interval offset is numeric.

**Years:** Specifies that the interval offset is years.

**Months:** Specifies that the interval offset is months.

**Weeks:** Specifies that the interval offset is weeks.

**Days:** Specifies that the interval offset is days.

**Hours:** Specifies that the interval offset is hours.

**Minutes:** Specifies that the interval offset is minutes.

**Seconds:** Specifies that the interval offset is seconds.

**Milliseconds:** Specifies that the interval offset is milliseconds.

If the **ChartGridLines.IntervalOffsetType** element is not present, its value is interpreted as "Default".

Following is the parent element of the **ChartGridLines.IntervalOffsetType** element.

Parent elements
<a href="#">ChartGridLines</a>

The following is the XML Schema definition of the **ChartGridLines.IntervalOffsetType** element.

```
<xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
```

### 2.123.5 ChartGridLines.IntervalType

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartGridLines.IntervalType** element specifies the unit to use for the [ChartGridLines.Interval](#) element. The **ChartGridLines.IntervalType** element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies that the value of [ChartAxis.IntervalType](#) is used.

**Auto:** Specifies that the interval unit is determined based on the data plotted against the axis.

**Number:** Specifies that the interval is numeric.

**Years:** Specifies that the interval is years.

**Months:** Specifies that the interval is months.

**Weeks:** Specifies that the interval is weeks.

**Days:** Specifies that the interval is days.

**Hours:** Specifies that the interval is hours.

**Minutes:** Specifies that the interval is minutes.

**Seconds:** Specifies that the interval is seconds.

**Milliseconds:** Specifies that the interval is milliseconds.

If the **ChartGridLines.IntervalType** element is not present, its value is interpreted as "Default".

Following is the parent element of the **ChartGridLines.IntervalType** element.

Parent elements
<a href="#">ChartGridLines</a>

The following is the XML Schema definition of the **ChartGridLines.IntervalType** element.

```
<xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
```

### 2.123.6 ChartGridLines.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartGridLines.Style** element specifies the line style properties for gridlines. This element is optional. This element is of type [Style](#).

Following is the parent element of the **ChartGridLines.Style** element.

Parent elements
<a href="#">ChartGridLines</a>

The following is the XML Schema definition of the **ChartGridLines.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.124 ChartStripLines

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartStripLines** element specifies an ordered list of custom **strip lines** for a [ChartAxis](#). The [ChartGridLines](#) element MUST contain at least one [ChartStripLine](#) element.

The following are the parent and child elements of the **ChartStripLines** element.

Parent elements
ChartAxis

Child elements
<a href="#">ChartStripLines.ChartStripLine</a>

The following is the XML Schema definition of the **ChartStripLines** element in RDL 2008/01.

```
<xsd:complexType name="ChartStripLinesType">  
  <xsd:sequence>  
    <xsd:element name="ChartStripLine" type="ChartStripLineType"  
      minOccurs="1" maxOccurs="unbounded" />  
  </xsd:sequence>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartStripLines** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartStripLinesType">  
  <xsd:sequence>
```

```

    <xsd:element name="ChartStripLine" type="ChartStripLineType"
                maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.124.1 ChartStripLines.ChartStripLine

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartStripLines.ChartStripLine** element specifies a custom strip line for a [ChartAxis](#). The **ChartStripLines.ChartStripLine** element MUST be specified at least once. This element is of type [ChartStripLine](#).

Following is the parent element of the **ChartStripLines.ChartStripLine** element.

Parent elements
<a href="#">ChartStripLines</a>

The following is the XML Schema definition of the **ChartStripLines.ChartStripLine** element.

```

<xsd:element name="ChartStripLine" type="ChartStripLineType"
                maxOccurs="unbounded" />

```

### 2.125 ChartStripLine

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartStripLine** element specifies a custom **strip line** for a [ChartAxis](#).

The following are the parent and child elements of the **ChartStripLine** element.

Parent elements
<a href="#">ChartStripLines</a>

Child elements
<a href="#">ChartStripLine.ActionInfo</a>
<a href="#">ChartStripLine.Interval</a>
<a href="#">ChartStripLine.IntervalOffset</a>
<a href="#">ChartStripLine.IntervalOffsetType</a>
<a href="#">ChartStripLine.IntervalType</a>
<a href="#">ChartStripLine.StripWidth</a>
<a href="#">ChartStripLine.StripWidthType</a>
<a href="#">ChartStripLine.Style</a>
<a href="#">ChartStripLine.TextOrientation</a>

Child elements
<a href="#">ChartStripLine.Title</a>
<a href="#">ChartStripLine.TitleAngle</a>
<a href="#">ChartStripLine.ToolTip</a>

The following is the XML Schema definition of the **ChartStripLine** element in RDL 2008/01.

```
<xsd:complexType name="ChartStripLineType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Title" type="xsd:string" minOccurs="0" />
    <xsd:element name="TitleAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:element name="StripWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="StripWidthType" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartStripLine** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartStripLineType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Title" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:element name="StripWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="StripWidthType" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.125.1 ChartStripLine.ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartStripLine.ActionInfo** element specifies actions that are associated with a strip line. This element is optional. This element is of type [ActionInfo](#).

Following is the parent element of the **ChartStripLine.ActionInfo** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartStripLine</a>
--------------------------------

The following is the XML Schema definition of the **ChartStripLine.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

## 2.125.2 ChartStripLine.Interval

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartStripLine.Interval** element specifies the size of a strip line interval.

The **ChartStripLine.Interval** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element specifies a value of 0, its value is determined by the axis. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **ChartStripLine.Interval** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartStripLine</a>
--------------------------------

The following is the XML Schema definition of the **ChartStripLine.Interval** element.

```
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
```

## 2.125.3 ChartStripLine.IntervalOffset

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartStripLine.IntervalOffset** element specifies the offset from the previous strip line. If this is the first strip line, the offset is interpreted as the offset from the value of **ChartAxis.Minimum** element. This element is optional.

If the **ChartStripLine.IntervalOffset** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element specifies a value of 0, the value of the **ChartAxis.IntervalOffset** element is used as the value of the **ChartStripLine.IntervalOffset** element.

If the **ChartStripLine.IntervalOffset** element is not present, its value is interpreted as 0.

<b>Parent elements</b>
------------------------

<a href="#">ChartStripLine</a>
--------------------------------

The following is the XML Schema definition of the **ChartStripLine.IntervalOffset** element.

```
<xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
```

## 2.125.4 ChartStripLine.IntervalOffsetType

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartStripLine.IntervalOffsetType** element specifies the unit to use for the [ChartStripLine.IntervalOffset](#) element. The **ChartStripLine.IntervalOffsetType** element is optional.

If the **ChartStripLine.IntervalOffsetType** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Auto:** Specifies that the strip line interval offset unit is determined based on the data plotted against the axis.

**Number:** Specifies that the strip line interval offset is numeric.

**Years:** Specifies that the strip line interval offset is years.

**Months:** Specifies that the strip line interval offset is months.

**Weeks:** Specifies that the strip line interval offset is weeks.

**Days:** Specifies that the strip line interval offset is days.

**Hours:** Specifies that the strip line interval offset is hours.

**Minutes:** Specifies that the strip line interval offset is minutes.

**Seconds:** Specifies that the strip line interval offset is seconds.

**Milliseconds:** Specifies that the strip line interval offset is milliseconds.

If the **ChartStripLine.IntervalOffsetType** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartStripLine.IntervalOffsetType** element.

Parent elements
<a href="#">ChartStripLine</a>

The following is the XML Schema definition of the **ChartStripLine.IntervalOffsetType** element.

```
<xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
```

## 2.125.5 ChartStripLine.IntervalType

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartStripLine.IntervalType** element specifies the unit to use for the [ChartStripLine.Interval](#) element. The **ChartStripLine.IntervalType** element is optional.

If the **ChartStripLine.IntervalType** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Auto:** Specifies that the strip line interval unit is determined based on the data plotted against the axis.

**Number:** Specifies that the strip line interval is numeric.

**Years:** Specifies that the strip line interval is years.

**Months:** Specifies that the strip line interval is months.

**Weeks:** Specifies that the strip line interval is weeks.

**Days:** Specifies that the strip line interval is days.

**Hours:** Specifies that the strip line interval is hours.

**Minutes:** Specifies that the strip line interval is minutes.

**Seconds:** Specifies that the strip line interval is seconds.

**Milliseconds:** Specifies that the strip line interval is milliseconds.

If the **ChartStripLine.IntervalType** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartStripLine.IntervalType** element.

Parent elements
<a href="#">ChartStripLine</a>

The following is the XML Schema definition of the **ChartStripLine.IntervalType** element.

```
<xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
```

## 2.125.6 ChartStripLine.StripWidth

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartStripLine.StripWidth** element specifies the width of a strip line. The **ChartStripLine.StripWidth** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **ChartStripLine.StripWidth** element.

Parent elements
<a href="#">ChartStripLine</a>

The following is the XML Schema definition of the **ChartStripLine.StripWidth** element.

```
<xsd:element name="StripWidth" type="xsd:string" minOccurs="0" />
```

## 2.125.7 ChartStripLine.StripWidthType

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartStripLine.StripWidthType** element specifies the unit to use for the [ChartStripLine.StripWidth](#) element. The **ChartStripLine.StripWidthType** element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:



**Auto:** Specifies that the strip line width unit is determined based on the data plotted against the axis.

**Number:** Specifies that the strip line width is numeric.

**Years:** Specifies that the strip line width is years.

**Months:** Specifies that the strip line width is months.

**Weeks:** Specifies that the strip line width is weeks.

**Days:** Specifies that the strip line width is days.

**Hours:** Specifies that the strip line width is hours.

**Minutes:** Specifies that the strip line width is minutes.

**Seconds:** Specifies that the strip line width is seconds.

**Milliseconds:** Specifies that the strip line width is milliseconds.

If the **ChartStripLine.StripWidthType** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartStripLine.StripWidthType** element.

Parent elements
<a href="#">ChartStripLine</a>

The following is the XML Schema definition of the **ChartStripLine.StripWidthType** element.

```
<xsd:element name="StripWidthType" type="xsd:string" minOccurs="0" />
```

## 2.125.8 ChartStripLine.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartStripLine.Style** element specifies the style properties for a strip line. This element is optional. This element is of type [Style](#).

Following is the parent element of the **ChartStripLine.Style** element.

Parent elements
<a href="#">ChartStripLine</a>

The following is the XML Schema definition of the **ChartStripLine.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.125.9 ChartStripLine.TextOrientation

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartStripLine.TextOrientation** element specifies the orientation of the title text for a strip line. This element is optional.

If the **ChartStripLine.TextOrientation** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Auto**: Specifies whether the orientation is selected automatically based on context; for example, "Rotated270" is selected for titles that are docked on the left.

**Horizontal**: Specifies horizontal text.

**Rotated90**: Specifies vertical text, rotated 90 degrees.

**Rotated270**: Specifies vertical text, rotated 270 degrees.

**Stacked**: Specifies vertical text with no character rotation.

If the **ChartStripLine.TextOrientation** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartStripLine.TextOrientation** element.

Parent elements
<a href="#">ChartStripLine</a>

The following is the XML Schema definition of the **ChartStripLine.TextOrientation** element.

```
<xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
```

## 2.125.10 ChartStripLine.Title

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartStripLine.Title** element specifies the title for a strip line. The **ChartStripLine.Title** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **ChartStripLine.Title** element.

Parent elements
<a href="#">ChartStripLine</a>

The following is the XML Schema definition of the **ChartStripLine.Title** element.

```
<xsd:element name="Title" type="xsd:string" minOccurs="0" />
```

## 2.125.11 ChartStripLine.TitleAngle

*Applies to [RDL 2008/01](#)*

The **ChartStripLine.TitleAngle** element is ignored.

Following is the parent element of the **ChartStripLine.TitleAngle** element.

<b>Parent elements</b>
<a href="#">ChartStripLine</a>

The following is the XML Schema definition of the **ChartStripLine.TitleAngle** element.

```
<xsd:element name="TitleAngle" type="xsd:string" minOccurs="0" />
```

### 2.125.12 ChartStripLine.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartStripLine.ToolTip** element specifies the tooltip to display for a strip line. The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **ChartStripLine.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **ChartStripLine.ToolTip** element.

<b>Parent elements</b>
<a href="#">ChartStripLine</a>

The following is the XML Schema definition of the **ChartStripLine.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

### 2.126 ChartTickMarks

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartTickMarks** element specifies tick marks along a [ChartAxis](#).

The following are the parent and child elements of the **ChartTickMarks** element.

<b>Parent elements</b>
<a href="#">ChartAxis</a>

<b>Child elements</b>
<a href="#">ChartTickMarks.Enabled</a>
<a href="#">ChartTickMarks.Interval</a>
<a href="#">ChartTickMarks.IntervalOffset</a>
<a href="#">ChartTickMarks.IntervalOffsetType</a>
<a href="#">ChartTickMarks.IntervalType</a>
<a href="#">ChartTickMarks.Length</a>

<b>Child elements</b>
<a href="#">ChartTickMarks.Style</a>
<a href="#">ChartTickMarks.Type</a>

The following is the XML Schema definition of the **ChartTickMarks** element in RDL 2008/01.

```
<xsd:complexType name="ChartTickMarksType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartTickMarks** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartTickMarksType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.126.1 **ChartTickMarks.Enabled**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTickMarks.Enabled** element specifies whether tick marks along a [ChartAxis](#) are shown. This element is optional. If the **ChartTickMarks.Enabled** element is present, its value **MUST** be a [String](#) ([[XMLSCHEMA2/2](#)] section 3.2.1) or an expression that evaluates to a **String**. The value of this element **MUST** be one of the following or an expression that evaluates to one of the following:

**True:** Specifies that tick marks appear.

**False:** Specifies that tick marks do not appear.

**Auto:** Specifies that this value is interpreted as true for [ChartAxis.ChartMajorTickMarks](#) and false for [ChartAxis.ChartMinorTickMarks](#).

If the **ChartTickMarks.Enabled** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartTickMarks.Enabled** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartTickMarks</a>
--------------------------------

The following is the XML Schema definition of the **ChartTickMarks.Enabled** element.

```
<xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
```

## 2.126.2 ChartTickMarks.Interval

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartTickMarks.Interval** element specifies the interval between tick marks on a [ChartAxis](#). This element is optional. If the **ChartTickMarks.Interval** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

If the value of the **ChartTickMarks.Interval** element is specified as 0, its value is determined by the axis. If the value of this element is specified as 0 or evaluates to 0, the value of [ChartAxis.Interval](#) will be used (or its value will be determined by the value of **ChartAxis.Interval**).

Following is the parent element of the **ChartTickMarks.Interval** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartTickMarks</a>
--------------------------------

The following is the XML Schema definition of the **ChartTickMarks.Interval** element.

```
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
```

## 2.126.3 ChartTickMarks.IntervalOffset

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartTickMarks.IntervalOffset** element specifies the offset for the first tick mark from the axis minimum. This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

If the value of the **ChartTickMarks.IntervalOffset** element is specified as 0, its value is determined by the axis. If the value of this element is specified as 0 or evaluates to 0, the value of [ChartAxis.IntervalOffset](#) will be used.

Following is the parent element of the **ChartTickMarks.IntervalOffset** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartTickMarks</a>
--------------------------------

The following is the XML Schema definition of the **ChartTickMarks.IntervalOffset** element.

```
<xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
```

## 2.126.4 ChartTickMarks.IntervalOffsetType

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTickMarks.IntervalOffsetType** element specifies the unit to use for the [ChartTickMarks.IntervalOffset](#) element. This element is optional. If the **ChartTickMarks.IntervalOffsetType** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies that [ChartAxis.IntervalOffsetType](#) is used.

**Auto:** Specifies that the interval offset unit is determined automatically based on the data plotted against the axis.

**Number:** Specifies that the interval offset is numeric.

**Years:** Specifies that the interval offset is years.

**Months:** Specifies that the interval offset is months.

**Weeks:** Specifies that the interval offset is weeks.

**Days:** Specifies that the interval offset is days.

**Hours:** Specifies that the interval offset is hours.

**Minutes:** Specifies that the interval offset is minutes.

**Seconds:** Specifies that the interval offset is seconds.

**Milliseconds:** Specifies that the interval offset is milliseconds.

If the **ChartTickMarks.IntervalOffsetType** element is not present, its value is interpreted as "Default".

Following is the parent element of the **ChartTickMarks.IntervalOffsetType** element.

<b>Parent elements</b>
<a href="#">ChartTickMarks</a>

The following is the XML Schema definition of the **ChartTickMarks.IntervalOffsetType** element.

```
<xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
```

## 2.126.5 ChartTickMarks.IntervalType

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTickMarks.IntervalType** element specifies the unit to use for the [ChartTickMarks.Interval](#) element. This element is optional. If the **ChartTickMarks.IntervalType** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies that [ChartAxis.IntervalType](#) is used.

**Auto:** Specifies that the interval unit is determined based on the data plotted against the axis.

**Number:** Specifies that interval is numeric.

**Years:** Specifies that the interval is years.

**Months:** Specifies that the interval is months.

**Weeks:** Specifies that the interval is weeks.

**Days:** Specifies that the interval is days.

**Hours:** Specifies that the interval is hours.

**Minutes:** Specifies that the interval is minutes.

**Seconds:** Specifies that the interval is seconds.

**Milliseconds:** Specifies that the interval is milliseconds.

If the **ChartTickMarks.IntervalType** element is not present, its value is interpreted as "Default".

Following is the parent element of the **ChartTickMarks.IntervalType** element.

Parent elements
<a href="#">ChartTickMarks</a>

The following is the XML Schema definition of the **ChartTickMarks.IntervalType** element.

```
<xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
```

## 2.126.6 ChartTickMarks.Length

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTickMarks.Length** element specifies the length of a tick mark as a percentage of the size of the chart. This element is optional. If the **ChartTickMarks.Length** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 1.

Following is the parent element of the **ChartTickMarks.Length** element.

Parent elements
<a href="#">ChartTickMarks</a>

The following is the XML Schema definition of the **ChartTickMarks.Length** element.

```
<xsd:element name="Length" type="xsd:string" minOccurs="0" />
```

## 2.126.7 ChartTickMarks.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTickMarks.Style** element specifies the line style properties for a tick mark. This element is optional. The **ChartTickMarks.Style** element is of type [Style](#).

Following is the parent element of the **ChartTickMarks.Style** element.

<b>Parent elements</b>
<a href="#">ChartTickMarks</a>

The following is the XML Schema definition of the **ChartTickMarks.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.126.8 ChartTickMarks.Type

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartTickMarks.Type** element specifies the type of a tick mark. This element is optional. If the **ChartTickMarks.Type** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Outside:** Specifies that tick marks appear outside the axis.

**Inside:** Specifies that tick marks appear inside the axis.

**Cross:** Specifies that tick marks appear across the axis.

**None:** Specifies that tick marks do not appear.

If the **ChartTickMarks.Type** element is not present, its value is interpreted as "Outside".

Following is the parent element of the **ChartTickMarks.Type** element.

<b>Parent elements</b>
<a href="#">ChartTickMarks</a>

The following is the XML Schema definition of the **ChartTickMarks.Type** element.

```
<xsd:element name="Type" type="xsd:string" minOccurs="0" />
```

## 2.127 ChartElementPosition

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartElementPosition** element specifies the position at which to draw a **chart item**, such as [ChartArea](#), [ChartLegend](#), or [ChartTitle](#).

The following are the parent and child elements of the **ChartElementPosition** element.

<b>Parent elements</b>
ChartArea
ChartLegend
ChartTitle



Child elements
<a href="#">ChartElementPosition.Height</a>
<a href="#">ChartElementPosition.Left</a>
<a href="#">ChartElementPosition.Top</a>
<a href="#">ChartElementPosition.Width</a>

The following is the XML Schema definition of the **ChartElementPosition** element in RDL 2008/01.

```
<xsd:complexType name="ChartElementPositionType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartElementPosition** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartElementPositionType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.127.1 ChartElementPosition.Height

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartElementPosition.Height** element specifies the height of a **chart item** as a percentage of the height of its containing object. This element is optional.

If the **ChartElementPosition.Height** element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 100 minus the **Float** value of [ChartElementPosition.Top](#). The value of the **ChartElementPosition.Height** element MUST be greater than or equal to 0 and less than or equal to 100.

Following is the parent element of the **ChartElementPosition.Height** element.

Parent elements
<a href="#">ChartElementPosition</a>

The following is the XML Schema definition of the **ChartElementPosition.Height** element.

```
<xsd:element name="Height" type="xsd:string" minOccurs="0" />
```

### 2.127.2 ChartElementPosition.Left

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartElementPosition.Left** element specifies the distance of a chart item from the left of the containing object as a percentage of the width of the container. This element is optional.

If the **ChartElementPosition.Left** element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0. The value of this element MUST be greater than or equal to 0 and less than or equal to 100.

Following is the parent element of the **ChartElementPosition.Left** element.

Parent elements
<a href="#">ChartElementPosition</a>

The following is the XML Schema definition of the **ChartElementPosition.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

### 2.127.3 ChartElementPosition.Top

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartElementPosition.Top** element specifies the distance of a chart item from the top of the containing object as a percentage of the height of the container. This element is optional.

If the **ChartElementPosition.Top** element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0. The value of this element MUST be greater than or equal to 0 and less than or equal to 100.

Following is the parent element of the **ChartElementPosition.Top** element.

Parent elements
<a href="#">ChartElementPosition</a>

The following is the XML Schema definition of the **ChartElementPosition.Top** element.

```
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
```

### 2.127.4 ChartElementPosition.Width

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartElementPosition.Width** element specifies the width of a chart item as a percentage the width of its containing object. This element is optional.

If the **ChartElementPosition.Width** element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 100 minus the value of [ChartElementPosition.Left](#). The value of the **ChartElementPosition.Width** element MUST be greater than or equal to 0 and less than or equal to 100.

Following is the parent element of the **ChartElementPosition.Width** element.

Parent elements
<a href="#">ChartElementPosition</a>

The following is the XML Schema definition of the **ChartElementPosition.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
```

## 2.128 ChartThreeDProperties

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartThreeDProperties** element specifies the properties of a 3D layout.

The following are the parent and child elements of the **ChartThreeDProperties** element.

Parent elements
<a href="#">ChartArea</a>

Child elements
<a href="#">ChartThreeDProperties.Clustered</a>
<a href="#">ChartThreeDProperties.DepthRatio</a>
<a href="#">ChartThreeDProperties.Enabled</a>
<a href="#">ChartThreeDProperties.GapDepth</a>
<a href="#">ChartThreeDProperties.Inclination</a>
<a href="#">ChartThreeDProperties.Perspective</a>
<a href="#">ChartThreeDProperties.ProjectionMode</a>
<a href="#">ChartThreeDProperties.Rotation</a>
<a href="#">ChartThreeDProperties.Shading</a>
<a href="#">ChartThreeDProperties.WallThickness</a>

The following is the XML Schema definition of the **ChartThreeDProperties** element in RDL 2008/01.

```
<xsd:complexType name="ChartThreeDPropertiesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="ProjectionMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="Rotation" type="xsd:string" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

```

<xsd:element name="Inclination" type="xsd:string" minOccurs="0" />
<xsd:element name="Perspective" type="xsd:string" minOccurs="0" />
<xsd:element name="DepthRatio" type="xsd:string" minOccurs="0" />
<xsd:element name="Shading" type="xsd:string" minOccurs="0" />
<xsd:element name="GapDepth" type="xsd:string" minOccurs="0" />
<xsd:element name="WallThickness" type="xsd:string" minOccurs="0" />
<xsd:element name="Clustered" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **ChartThreeDProperties** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="ChartThreeDPropertiesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="ProjectionMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="Rotation" type="xsd:string" minOccurs="0" />
    <xsd:element name="Inclination" type="xsd:string" minOccurs="0" />
    <xsd:element name="Perspective" type="xsd:string" minOccurs="0" />
    <xsd:element name="DepthRatio" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shading" type="xsd:string" minOccurs="0" />
    <xsd:element name="GapDepth" type="xsd:string" minOccurs="0" />
    <xsd:element name="WallThickness" type="xsd:string" minOccurs="0" />
    <xsd:element name="Clustered" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.128.1 ChartThreeDProperties.Clustered

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartThreeDProperties.Clustered** element specifies whether data series are clustered (displayed along distinct rows). This element is optional. Data series MUST NOT be displayed as clustered if the value of the [ChartSeries.Type](#) element is not "Bar" or "Column".

If the **ChartThreeDProperties.Clustered** element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartThreeDProperties.Clustered** element.

Parent elements
<a href="#">ChartThreeDProperties</a>

The following is the XML Schema definition of the **ChartThreeDProperties.Clustered** element.

```

<xsd:element name="Clustered" type="xsd:string" minOccurs="0" />

```

### 2.128.2 ChartThreeDProperties.DepthRatio

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartThreeDProperties.DepthRatio** element specifies the ratio between depth and width as a percentage. This element is optional.

If the **ChartThreeDProperties.DepthRatio** element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 100. The value of this element MUST be greater than or equal to 0. <26>

Following is the parent element of the **ChartThreeDProperties.DepthRatio** element.

Parent elements
<a href="#">ChartThreeDProperties</a>

The following is the XML Schema definition of the **ChartThreeDProperties.DepthRatio** element.

```
<xsd:element name="DepthRatio" type="xsd:string" minOccurs="0" />
```

### 2.128.3 ChartThreeDProperties.Enabled

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartThreeDProperties.Enabled** element specifies whether a [Chart](#) is displayed in 3D. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartThreeDProperties.Enabled** element.

Parent elements
<a href="#">ChartThreeDProperties</a>

The following is the XML Schema definition of the **ChartThreeDProperties.Enabled** element.

```
<xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
```

### 2.128.4 ChartThreeDProperties.GapDepth

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartThreeDProperties.GapDepth** element specifies the depth gap between 3D bars and columns as a percentage of the distance between **data points** in one row. This element is optional.

If the **ChartThreeDProperties.GapDepth** element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 100. The value of this element MUST be greater than or equal to 0 and less than or equal to 2147483647.

Following is the parent element of the **ChartThreeDProperties.GapDepth** element.

Parent elements
<a href="#">ChartThreeDProperties</a>

The following is the XML Schema definition of the **ChartThreeDProperties.GapDepth** element.

```
<xsd:element name="GapDepth" type="xsd:string" minOccurs="0" />
```

### 2.128.5 ChartThreeDProperties.Inclination

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartThreeDProperties.Inclination** element specifies the inclination angle. If the **ChartThreeDProperties.Inclination** element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 30. The value of this element MUST be greater than or equal to 0 and less than or equal to 90.

Following is the parent element of the **ChartThreeDProperties.Inclination** element.

Parent elements
<a href="#">ChartThreeDProperties</a>

The following is the XML Schema definition of the **ChartThreeDProperties.Inclination** element.

```
<xsd:element name="Inclination" type="xsd:string" minOccurs="0" />
```

### 2.128.6 ChartThreeDProperties.Perspective

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartThreeDProperties.Perspective** element specifies the percentage of perspective in a 3D chart. This element is optional. This element is ignored if the value of the **ChartThreeDProperties.ProjectionMode** element is not "Perspective."

If the **ChartThreeDProperties.Perspective** element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 0. The value of this element MUST be greater than or equal to 0 and less than or equal to 100.

Following is the parent element of the **ChartThreeDProperties.Perspective** element.

Parent elements
<a href="#">ChartThreeDProperties</a>

The following is the XML Schema definition of the **ChartThreeDProperties.Perspective** element.

```
<xsd:element name="Perspective" type="xsd:string" minOccurs="0" />
```

### 2.128.7 ChartThreeDProperties.ProjectionMode

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartThreeDProperties.ProjectionMode** element specifies the projection mode that is used for a 3D rendering. This element is optional.

If the **ChartThreeDProperties.ProjectionMode** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Oblique:** Specifies that an oblique projection is used.

**Perspective:** Specifies that a perspective projection is used.

If this element is not present, its value is interpreted as "Oblique".

Following is the parent element of the **ChartThreeDProperties.ProjectionMode** element.

Parent elements
<a href="#">ChartThreeDProperties</a>

The following is the XML Schema definition of the **ChartThreeDProperties.ProjectionMode** element.

```
<xsd:element name="ProjectionMode" type="xsd:string" minOccurs="0" />
```

## 2.128.8 ChartThreeDProperties.Rotation

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartThreeDProperties.Rotation** element specifies the rotation angle. This element is optional.

If the **ChartThreeDProperties.Rotation** element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 30. [<27>](#)

Following is the parent element of the **ChartThreeDProperties.Rotation** element.

Parent elements
<a href="#">ChartThreeDProperties</a>

The following is the XML Schema definition of the **ChartThreeDProperties.Rotation** element.

```
<xsd:element name="Rotation" type="xsd:string" minOccurs="0" />
```

## 2.128.9 ChartThreeDProperties.Shading

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartThreeDProperties.Shading** element specifies the type of 3D shading. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**.

The value of the **ChartThreeDProperties.Shading** element MUST be one of the following:

**Real:** Specifies realistic shading.

**Simple:** Specifies simplified shading.

**None:** Specifies no shading.

If the **ChartThreeDProperties.Shading** element is not present, its value is interpreted as "Real".

Following is the parent element of the **ChartThreeDProperties.Shading** element.

Parent elements
<a href="#">ChartThreeDProperties</a>

The following is the XML Schema definition of the **ChartThreeDProperties.Shading** element.

```
<xsd:element name="Shading" type="xsd:string" minOccurs="0" />
```

### 2.128.10 ChartThreeDProperties.WallThickness

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartThreeDProperties.WallThickness** element specifies the thickness of outer walls as a percentage. This element is optional.

If the **ChartThreeDProperties.WallThickness** element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 7. The value of this element MUST be greater than or equal to 0. <28>

Following is the parent element of the **ChartThreeDProperties.WallThickness** element.

Parent elements
<a href="#">ChartThreeDProperties</a>

The following is the XML Schema definition of the **ChartThreeDProperties.WallThickness** element.

```
<xsd:element name="WallThickness" type="xsd:string" minOccurs="0" />
```

### 2.129 ChartValueAxes

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartValueAxes** element defines a list of value (Y) axes. This element specifies the collection of [ChartAxis](#) instances as value (Y) axes for a [ChartArea](#).

The **ChartValueAxes** element SHOULD NOT contain more than one **ChartAxis** that has the [ChartAxis.Location](#) element set to "Default". Additionally, the **ChartValueAxes** element SHOULD NOT contain more than one chart axis that has the **ChartAxis.Location** element set to "Opposite".

The following are the parent and child elements of the **ChartValueAxes** element.

Parent elements
ChartArea



<b>Child elements</b>
-----------------------

<a href="#">ChartValueAxes.ChartAxis</a>
--

The following is the XML Schema definition of the **ChartValueAxes** element in RDL 2008/01.

```
<xsd:complexType name="ChartValueAxesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartAxis" type="ChartAxisType" minOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartValueAxes** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartValueAxesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartAxis" type="ChartAxisType" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.129.1 ChartValueAxes.ChartAxis

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartValueAxes.ChartAxis** element specifies a [ChartAxis](#) for a [ChartArea](#). The **ChartValueAxes.ChartAxis** element MUST be specified at least once. The **ChartValueAxes.ChartAxis** element is of type **ChartAxis**.

Following is the parent element of the **ChartValueAxes.ChartAxis** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartValueAxes</a>
--------------------------------

The following is the XML Schema definition of the **ChartValueAxes.ChartAxis** element.

```
<xsd:element name="ChartAxis" type="ChartAxisType" minOccurs="1" />
```

### 2.130 ChartBorderSkin

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartBorderSkin** element specifies the appearance of the **ChartBorderSkin** around a [PlotArea](#).

The following are the parent and child elements of the **ChartBorderSkin** element.

<b>Parent elements</b>
------------------------

<a href="#">Chart</a>
-----------------------

<b>Child elements</b>
<a href="#">ChartBorderSkin.ChartBorderSkinType</a>
<a href="#">ChartBorderSkin.Style</a>

The following is the XML Schema definition of the **ChartBorderSkin** element in RDL 2008/01.

```
<xsd:complexType name="ChartBorderSkinType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartBorderSkinType" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartBorderSkin** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartBorderSkinType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartBorderSkinType" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
</xsd:complexType>
```

### 2.130.1 **ChartBorderSkin.ChartBorderSkinType**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartBorderSkin.ChartBorderSkinType** element specifies the **border skin** type for a [Chart](#). This element is optional. If the **ChartBorderSkin.ChartBorderSkinType** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**None**: Specifies no border skin.

**Emboss**: Specifies an embossed border skin.

**Raised**: Specifies a raised border skin.

**Sunken**: Specifies a sunken border skin.

**FrameThin1**: Specifies the FrameThin1 border skin.

**FrameThin2**: Specifies the FrameThin2 border skin.

**FrameThin3**: Specifies the FrameThin3 border skin.

**FrameThin4**: Specifies the FrameThin4 border skin.

**FrameThin5**: Specifies the FrameThin5 border skin.

**FrameThin6**: Specifies the FrameThin6 border skin.

**FrameTitle1**: Specifies the FrameTitle1 border skin.

**FrameTitle2:** Specifies the FrameTitle2 border skin.

**FrameTitle3:** Specifies the FrameTitle3 border skin.

**FrameTitle4:** Specifies the FrameTitle4 border skin.

**FrameTitle5:** Specifies the FrameTitle5 border skin.

**FrameTitle6:** Specifies the FrameTitle6 border skin.

**FrameTitle7:** Specifies the FrameTitle7 border skin.

**FrameTitle8:** Specifies the FrameTitle8 border skin.

If the **ChartBorderSkin.ChartBorderSkinType** element is not present, its value is interpreted as "None".

The following is the parent element of the **ChartBorderSkin.ChartBorderSkinType** element.

Parent elements
<a href="#">ChartBorderSkin</a>

The following is the XML Schema definition of the **ChartBorderSkin.ChartBorderSkinType** element.

```
<xsd:element name="ChartBorderSkinType" type="xsd:string" minOccurs="0" />
```

## 2.130.2 ChartBorderSkin.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartBorderSkin.Style** element specifies the style properties of a **border skin** for a [Chart](#). This element is optional. This element is of type [Style](#).

Following is the parent element of the **ChartBorderSkin.Style** element.

Parent elements
<a href="#">ChartBorderSkin</a>

The following is the XML Schema definition of the **ChartBorderSkin.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.131 ChartCodeParameters

**Applies to** [RDL 2008/01](#)

The **ChartCodeParameters** element is ignored.

The following are the parent and child elements of the **ChartCodeParameters** element.

Parent elements
<a href="#">Chart</a>

<b>Child elements</b>
-----------------------

<a href="#">ChartCodeParameters.ChartCodeParameter</a>
--

The following is the XML Schema definition of the **ChartCodeParameters** element.

```
<xsd:complexType name="ChartCodeParametersType">
  <xsd:sequence>
    <xsd:element name="ChartCodeParameter" type="ChartCodeParameterType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.131.1 ChartCodeParameters.ChartCodeParameter

*Applies to [RDL 2008/01](#)*

The **ChartCodeParameters.ChartCodeParameter** element is ignored.

Following is the parent element of the **ChartCodeParameters.ChartCodeParameter** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartCodeParameters</a>
-------------------------------------

The following is the XML Schema definition of the **ChartCodeParameters.ChartCodeParameter** element.

```
<xsd:element name="ChartCodeParameter" type="ChartCodeParameterType"
  maxOccurs="unbounded" />
```

### 2.132 ChartCodeParameter

*Applies to [RDL 2008/01](#)*

The **ChartCodeParameter** element is ignored.

The following are the parent elements, attributes, and child elements of the **ChartCodeParameter** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartCodeParameters</a>
-------------------------------------

<b>Attributes</b>
-------------------

<a href="#">ChartCodeParameter.Name</a>
---

<b>Child elements</b>
-----------------------

<a href="#">ChartCodeParameter.Value</a>
--

The following is the XML Schema definition of the **ChartCodeParameter** element.

```
<xsd:complexType name="ChartCodeParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.132.1 ChartCodeParameter.Name

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartCodeParameter.Name** attribute is ignored.

Following is the parent element of the **ChartCodeParameter.Name** attribute.

<b>Parent elements</b>
------------------------

<a href="#">ChartCodeParameter</a>
------------------------------------

The following is the XML Schema definition of the **ChartCodeParameter.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.132.2 ChartCodeParameter.Value

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartCodeParameter.Value** element is ignored.

Following is the parent element of the **ChartCodeParameter.Value** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartCodeParameter</a>
------------------------------------

The following is the XML Schema definition of the **ChartCodeParameter.Value** element.

```
<xsd:element name="Value" type="xsd:string" />
```

## 2.133 ChartCustomPaletteColors

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartCustomPaletteColors** element specifies an ordered list of colors to use for the custom palette of a **Chart**. This element **MUST** contain at least one [ChartCustomPaletteColors.ChartCustomPaletteColor](#) element.

The following are the parent and child elements of the **ChartCustomPaletteColors** element.

Parent elements
<a href="#">Chart</a>

Child elements
ChartCustomPaletteColors.ChartCustomPaletteColor

The following is the XML Schema definition of the **ChartCustomPaletteColors** element in RDL 2008/01.

```
<xsd:complexType name="ChartCustomPaletteColorsType">
  <xsd:sequence>
    <xsd:element name="ChartCustomPaletteColor" type="xsd:string"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartCustomPaletteColors** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartCustomPaletteColorsType">
  <xsd:sequence>
    <xsd:element name="ChartCustomPaletteColor" type="xsd:string"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.133.1 ChartCustomPaletteColors.ChartCustomPaletteColor

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartCustomPaletteColors.ChartCustomPaletteColor** element specifies a color for the custom palette. This element **MUST** be specified at least once. The value of this element **MUST** be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

Following is the parent element of the **ChartCustomPaletteColors.ChartCustomPaletteColor** element.

Parent elements
<a href="#">ChartCustomPaletteColors</a>

The following is the XML Schema definition of the **ChartCustomPaletteColors.ChartCustomPaletteColor** element.

```
<xsd:element name="ChartCustomPaletteColor" type="xsd:string" maxOccurs="unbounded" />
```

## 2.134 ChartDerivedSeriesCollection

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDerivedSeriesCollection** element specifies an ordered list of [ChartDerivedSeries](#) elements.

The following are the parent and child elements of the **ChartDerivedSeriesCollection** element.

Parent elements
<a href="#">ChartData</a>

Child elements
<a href="#">ChartDerivedSeriesCollection.ChartDerivedSeries</a>

The following is the XML Schema definition of the **ChartDerivedSeriesCollection** element in RDL 2008/01.

```
<xsd:complexType name="ChartDerivedSeriesCollectionType">
  <xsd:sequence maxOccurs="unbounded" minOccurs="1">
    <xsd:element name="ChartDerivedSeries" type="ChartDerivedSeriesType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartDerivedSeriesCollection** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartDerivedSeriesCollectionType">
  <xsd:sequence maxOccurs="unbounded" minOccurs="1">
    <xsd:element name="ChartDerivedSeries" type="ChartDerivedSeriesType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.134.1 ChartDerivedSeriesCollection.ChartDerivedSeries

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDerivedSeriesCollection.ChartDerivedSeries** element specifies a [ChartDerivedSeries](#) within the [ChartDerivedSeriesCollection](#) for a [ChartData](#) element. The **ChartDerivedSeriesCollection.ChartDerivedSeries** element MUST be specified at least once for the **ChartDerivedSeriesCollection** element.

This element is of type **ChartDerivedSeries**.

Following is the parent element of the **ChartDerivedSeriesCollection.ChartDerivedSeries** element.

Parent elements
ChartDerivedSeriesCollection

The following is the XML Schema definition of the **ChartDerivedSeriesCollection.ChartDerivedSeries** element.

```
<xsd:element name="ChartDerivedSeries" type="ChartDerivedSeriesType" minOccurs="1"
maxOccurs="unbounded" />
```

## 2.135 ChartDerivedSeries

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDerivedSeries** element specifies a **derived series** that is calculated from a formula that is applied to another series.

The following are the parent and child elements of the **ChartDerivedSeries** element.

Parent elements
<a href="#">ChartDerivedSeriesCollection</a>

Child elements
<a href="#">ChartDerivedSeries.ChartFormulaParameters</a>
<a href="#">ChartDerivedSeries.ChartSeries</a>
<a href="#">ChartDerivedSeries.DerivedSeriesFormula</a>
<a href="#">ChartDerivedSeries.SourceChartSeriesName</a>

The following is the XML Schema definition of the **ChartDerivedSeries** element in RDL 2008/01.

```
<xsd:complexType name="ChartDerivedSeriesType">
  <xsd:choice minOccurs="3" maxOccurs="unbounded">
    <xsd:element name="ChartSeries" type="ChartSeriesType" minOccurs="1" />
    <xsd:element name="SourceChartSeriesName" type="xsd:string" minOccurs="1" />
    <xsd:element name="DerivedSeriesFormula" minOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="RuningTotal" />
          <xsd:enumeration value="RuningAverage" />
          <xsd:enumeration value="MoneyFlow" />
          <xsd:enumeration value="OnBalanceVolume" />
          <xsd:enumeration value="NegativeVoluneIndex" />
          <xsd:enumeration value="PositiveVolumeIndex" />
          <xsd:enumeration value="PriceVolumeTrend" />
          <xsd:enumeration value="AccumulationDistribution" />
          <xsd:enumeration value="Forecasting" />
          <xsd:enumeration value="StandardDeviation" />
          <xsd:enumeration value="AverageTruerance" />
          <xsd:enumeration value="EaseOfMovement" />
          <xsd:enumeration value="MassIndex" />
          <xsd:enumeration value="Performance" />
          <xsd:enumeration value="RateOfChange" />
          <xsd:enumeration value="RelativeStrengthIndex" />
          <xsd:enumeration value="TRIX" />
          <xsd:enumeration value="MACD" />
          <xsd:enumeration value="CommodityChannelIndex" />
          <xsd:enumeration value="TTestEqualVariances" />
          <xsd:enumeration value="TTestUnequalVariances" />
          <xsd:enumeration value="TTestPaired" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>
```



```

    <xsd:enumeration value="ZTest" />
    <xsd:enumeration value="FTest" />
    <xsd:enumeration value="Covariance" />
    <xsd:enumeration value="Correlation" />
    <xsd:enumeration value="Anova" />
    <xsd:enumeration value="TDistribution" />
    <xsd:enumeration value="FDistribution" />
    <xsd:enumeration value="NormalDistribution" />
    <xsd:enumeration value="InverseTDistribution" />
    <xsd:enumeration value="InverseFDistribution" />
    <xsd:enumeration value="InverseNormalDistribution" />
    <xsd:enumeration value="Mean" />
    <xsd:enumeration value="Variance" />
    <xsd:enumeration value="Median" />
    <xsd:enumeration value="BetaFunction" />
    <xsd:enumeration value="GammaFunction" />
    <xsd:enumeration value="MovingAverage" />
    <xsd:enumeration value="ExponentialMovingAverage" />
    <xsd:enumeration value="TriangularMovingAverage" />
    <xsd:enumeration value="WeightedMovingAverage" />
    <xsd:enumeration value="BollingerBands" />
    <xsd:enumeration value="MedianPrice" />
    <xsd:enumeration value="TypicalPrice" />
    <xsd:enumeration value="WeightedClose" />
    <xsd:enumeration value="Envelopes" />
    <xsd:enumeration value="StochasticIndicator" />
    <xsd:enumeration value="ChaikinOscillator" />
    <xsd:enumeration value="DetrendedPriceOscillator" />
    <xsd:enumeration value="VolatilityChankins" />
    <xsd:enumeration value="VolumeOscillator" />
    <xsd:enumeration value="WilliansR" />
  </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="ChartFormulaParameters" type="ChartFormulaParametersType"
  minOccurs="0" />
  <xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **ChartDerivedSeries** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="ChartDerivedSeriesType">
  <xsd:choice minOccurs="3" maxOccurs="unbounded">
    <xsd:element name="ChartSeries" type="ChartSeriesType" minOccurs="1" />
    <xsd:element name="SourceChartSeriesName" type="xsd:string" minOccurs="1" />
    <xsd:element name="DerivedSeriesFormula" minOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="StandardDeviation" />
          <xsd:enumeration value="Performance" />
          <xsd:enumeration value="RateOfChange" />
          <xsd:enumeration value="RelativeStrengthIndex" />
          <xsd:enumeration value="TRIX" />
          <xsd:enumeration value="MACD" />
          <xsd:enumeration value="Mean" />
          <xsd:enumeration value="Median" />
          <xsd:enumeration value="MovingAverage" />
          <xsd:enumeration value="ExponentialMovingAverage" />
          <xsd:enumeration value="TriangularMovingAverage" />
          <xsd:enumeration value="WeightedMovingAverage" />
          <xsd:enumeration value="BollingerBands" />
          <xsd:enumeration value="Envelopes" />
          <xsd:enumeration value="DetrendedPriceOscillator" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>

```

```

    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="ChartFormulaParameters" type="ChartFormulaParametersType"
    minOccurs="0" />
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.135.1 ChartDerivedSeries.ChartFormulaParameters

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDerivedSeries.ChartFormulaParameters** element specifies a set of [ChartFormulaParameter](#) instances for a formula. The **ChartDerivedSeries.ChartFormulaParameters** element is optional. This element is of type [ChartFormulaParameters](#).

Following is the parent element of the **ChartDerivedSeries.ChartFormulaParameters** element.

Parent elements
<a href="#">ChartDerivedSeries</a>

The following is the XML Schema definition of the **ChartDerivedSeries.ChartFormulaParameters** element.

```

<xsd:element name="ChartFormulaParameters" type="ChartFormulaParametersType"
  minOccurs="0" />

```

### 2.135.2 ChartDerivedSeries.ChartSeries

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDerivedSeries.ChartSeries** element specifies the series properties for a **derived series**. This element **MUST** be specified. This element is of type [ChartSeries](#).

Following is the parent element of the **ChartDerivedSeries.ChartSeries** element.

Parent elements
<a href="#">ChartDerivedSeries</a>

The following is the XML Schema definition of the **ChartDerivedSeries.ChartSeries** element.

```

<xsd:element name="ChartSeries" type="ChartSeriesType" minOccurs="1" />

```

### 2.135.3 ChartDerivedSeries.DerivedSeriesFormula

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDerivedSeries.DerivedSeriesFormula** element specifies the formula [\[DUNFORM\]](#) to apply to the data values from the [ChartSeries](#) that is specified by the [ChartDerivedSeries.SourceChartSeriesName](#) element. The

**ChartDerivedSeries.DerivedSeriesFormula** element MUST be specified. Its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1).

The value of this element MUST be one of the following:

**RuningTotal**: Specifies a running total formula.

**RuningAverage**: Specifies a running average formula.

**MoneyFlow**: Specifies a money flow formula.

**OnBalanceVolume**: Specifies an on-balance volume formula.

**NegativeVoluneIndex**: Specifies a negative volume index formula.

**PositiveVolumeIndex**: Specifies a positive volume index formula.

**PriceVolumeTrend**: Specifies a price volume trend formula.

**AccumulationDistribution**: Specifies an accumulation distribution formula.

**Forecasting**: Specifies a forecasting formula.

**StandardDeviation**: Specifies a standard deviation formula.

**AverageTruerance**: Specifies an average truerance formula.

**EaseOfMovement**: Specifies an ease of movement formula.

**MassIndex**: Specifies a mass index formula.

**Performance**: Specifies a performance formula.

**RateOfChange**: Specifies a rate of change formula.

**RelativeStrengthIndex**: Specifies a relative strength index formula.

**TRIX**: Specifies a TRIX formula.

**MACD**: Specifies an MACD formula.

**CommodityChannelIndex**: Specifies a commodity channel index formula.

**TTestEqualVariances**: Specifies a T test formula with equal variances.

**TTestUnequalVariances**: Specifies a T test formula with unequal variances.

**TTestPaired**: Specifies a T test formula with paired samples.

**ZTest**: Specifies a Z test formula.

**FTest**: Specifies an F test formula.

**Covariance**: Specifies a covariance formula.

**Correlation**: Specifies a correlation formula.

**Anova**: Specifies an ANOVA test formula.

**TDistribution**: Specifies a T distribution formula.

**FDistribution**: Specifies an F distribution formula.

**NormalDistribution:** Specifies a normal distribution formula.

**InverseTDistribution:** Specifies an inverse T distribution formula.

**InverseFDistribution:** Specifies an inverse F distribution formula.

**InverseNormalDistribution:** Specifies an inverse normal distribution formula.

**Mean:** Specifies a mean formula.

**Variance:** Specifies a variance formula.

**Median:** Specifies a median formula.

**BetaFunction:** Specifies a beta function formula.

**GammaFunction:** Specifies a gamma function formula.

**MovingAverage:** Specifies a moving average formula.

**ExponentialMovingAverage:** Specifies an exponential moving average formula.

**TriangularMovingAverage:** Specifies a triangular moving average formula.

**WeightedMovingAverage:** Specifies a weighted moving average formula.

**BollingerBands:** Specifies a Bollinger bands formula.

**MedianPrice:** Specifies a median price formula.

**TypicalPrice:** Specifies a typical price formula.

**WeightedClose:** Specifies a weighted close formula.

**Envelopes:** Specifies an envelope formula.

**StochasticIndicator:** Specifies a stochastic indicator formula.

**ChaikinOscillator:** Specifies a Chaikin oscillator formula.

**DetrendedPriceOscillator:** Specifies a detrended price oscillator formula.

**VolatilityChankins:** Specifies a volatility Chaikins formula.

**VolumeOscillator:** Specifies a volume oscillator formula.

**WilliansR:** Specifies a Williams %R formula.

Following is the parent element of the **ChartDerivedSeries.DerivedSeriesFormula** element.

<b>Parent elements</b>
<a href="#">ChartDerivedSeries</a>

The following is the XML Schema definition of the **ChartDerivedSeries.DerivedSeriesFormula** element in RDL 2008/01.

```
<xsd:element name="DerivedSeriesFormula" minOccurs="1">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="RuningTotal" />
      <xsd:enumeration value="RuningAverage" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

```

<xsd:enumeration value="MoneyFlow" />
<xsd:enumeration value="OnBalanceVolume" />
<xsd:enumeration value="NegativeVolumeIndex" />
<xsd:enumeration value="PositiveVolumeIndex" />
<xsd:enumeration value="PriceVolumeTrend" />
<xsd:enumeration value="AccumulationDistribution" />
<xsd:enumeration value="Forecasting" />
<xsd:enumeration value="StandardDeviation" />
<xsd:enumeration value="AverageTruerance" />
<xsd:enumeration value="EaseOfMovement" />
<xsd:enumeration value="MassIndex" />
<xsd:enumeration value="Performance" />
<xsd:enumeration value="RateOfChange" />
<xsd:enumeration value="RelativeStrengthIndex" />
<xsd:enumeration value="TRIX" />
<xsd:enumeration value="MACD" />
<xsd:enumeration value="CommodityChannelIndex" />
<xsd:enumeration value="TTestEqualVariances" />
<xsd:enumeration value="TTestUnequalVariances" />
<xsd:enumeration value="TTestPaired" />
<xsd:enumeration value="ZTest" />
<xsd:enumeration value="FTest" />
<xsd:enumeration value="Covariance" />
<xsd:enumeration value="Correlation" />
<xsd:enumeration value="Anova" />
<xsd:enumeration value="TDistribution" />
<xsd:enumeration value="FDistribution" />
<xsd:enumeration value="NormalDistribution" />
<xsd:enumeration value="InverseTDistribution" />
<xsd:enumeration value="InverseFDistribution" />
<xsd:enumeration value="InverseNormalDistribution" />
<xsd:enumeration value="Mean" />
<xsd:enumeration value="Variance" />
<xsd:enumeration value="Median" />
<xsd:enumeration value="BetaFunction" />
<xsd:enumeration value="GammaFunction" />
<xsd:enumeration value="MovingAverage" />
<xsd:enumeration value="ExponentialMovingAverage" />
<xsd:enumeration value="TriangularMovingAverage" />
<xsd:enumeration value="WeightedMovingAverage" />
<xsd:enumeration value="BollingerBands" />
<xsd:enumeration value="MedianPrice" />
<xsd:enumeration value="TypicalPrice" />
<xsd:enumeration value="WeightedClose" />
<xsd:enumeration value="Envelopes" />
<xsd:enumeration value="StochasticIndicator" />
<xsd:enumeration value="ChaikinOscillator" />
<xsd:enumeration value="DetrendedPriceOscillator" />
<xsd:enumeration value="VolatilityChankins" />
<xsd:enumeration value="VolumeOscillator" />
<xsd:enumeration value="WilliansR" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>

```

The following is the XML Schema definition of the **ChartDerivedSeries.DerivedSeriesFormula** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:element name="DerivedSeriesFormula" minOccurs="1">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="StandardDeviation" />
      <xsd:enumeration value="Performance" />
      <xsd:enumeration value="RateOfChange" />
      <xsd:enumeration value="RelativeStrengthIndex" />
      <xsd:enumeration value="TRIX" />
      <xsd:enumeration value="MACD" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

```

```

<xsd:enumeration value="Mean" />
<xsd:enumeration value="Median" />
<xsd:enumeration value="MovingAverage" />
<xsd:enumeration value="ExponentialMovingAverage" />
<xsd:enumeration value="TriangularMovingAverage" />
<xsd:enumeration value="WeightedMovingAverage" />
<xsd:enumeration value="BollingerBands" />
<xsd:enumeration value="Envelopes" />
<xsd:enumeration value="DetrendedPriceOscillator" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>

```

#### 2.135.4 ChartDerivedSeries.SourceChartSeriesName

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDerivedSeries.SourceChartSeriesName** element specifies the name of the [ChartSeries](#) from which to derive the series. This element **MUST** be specified, and its value **MUST** be the name of a **ChartSeries** within the same [Chart](#).

Following is the parent element of the **ChartDerivedSeries.SourceChartSeriesName** element.

Parent elements
<a href="#">ChartDerivedSeries</a>

The following is the XML Schema definition of the **ChartDerivedSeries.SourceChartSeriesName** element.

```
<xsd:element name="SourceChartSeriesName" type="xsd:string" minOccurs="1" />
```

#### 2.136 ChartFormulaParameters

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartFormulaParameters** element specifies an ordered list of [ChartFormulaParameter](#) elements. The **ChartFormulaParameters** element **MUST** have at least one **ChartFormulaParameter** child element specified.

The following are the parent and child elements of the **ChartFormulaParameters** element.

Parent elements
<a href="#">ChartDerivedSeries</a>

Child elements
<a href="#">ChartFormulaParameters.ChartFormulaParameter</a>

The following is the XML Schema definition of the **ChartFormulaParameters** element in RDL 2008/01.

```
<xsd:complexType name="ChartFormulaParametersType">
  <xsd:sequence>
```

```

    <xsd:element name="ChartFormulaParameter" type="ChartFormulaParameterType"
                maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **ChartFormulaParameters** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="ChartFormulaParametersType">
  <xsd:sequence>
    <xsd:element name="ChartFormulaParameter" type="ChartFormulaParameterType"
                maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.136.1 ChartFormulaParameters.ChartFormulaParameter

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartFormulaParameters.ChartFormulaParameter** element specifies a [ChartFormulaParameter](#) within the [ChartFormulaParameters](#) for a [ChartDerivedSeries](#). The **ChartFormulaParameters.ChartFormulaParameter** element MUST be specified at least once. This element is of type **ChartFormulaParameter**.

Following is the parent element of the **ChartFormulaParameters.ChartFormulaParameter** element.

Parent elements
<a href="#">ChartFormulaParameters</a>

The following is the XML Schema definition of the **ChartFormulaParameters.ChartFormulaParameter** element.

```

<xsd:element name="ChartFormulaParameter" type="ChartFormulaParameterType"
                maxOccurs="unbounded" />

```

### 2.137 ChartFormulaParameter

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartFormulaParameter** element specifies a parameter for a formula for a **derived series**. Depending on the value of the [ChartDerivedSeries.DerivedSeriesFormula](#) element, either the [ChartFormulaParameter.Value](#) or the [ChartFormulaParameter.Source](#) element MUST be specified.

The default value of the formula parameter is also dependent on the value of the **ChartDerivedSeries.DerivedSeriesFormula** element.

The following are the parent elements, attributes, and child elements of the **ChartFormulaParameter** element.

Parent elements
<a href="#">ChartFormulaParameters</a>

Attributes
<a href="#">ChartFormulaParameter.Name</a>

Child elements
ChartFormulaParameter.Source
ChartFormulaParameter.Value

The following is the XML Schema definition of the **ChartFormulaParameter** element in RDL 2008/01.

```
<xsd:complexType name="ChartFormulaParameterType">
  <xsd:choice minOccurs="1">
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Source" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartFormulaParameter** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartFormulaParameterType">
  <xsd:choice minOccurs="1">
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Source" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.137.1 ChartFormulaParameter.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartFormulaParameter.Name** attribute specifies the name of the [ChartFormulaParameter](#). This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [UTR15].

Following is the parent element of the **ChartFormulaParameter.Name** attribute.

Parent elements
ChartFormulaParameter

The following is the XML Schema definition of the **ChartFormulaParameter.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:string" use="required" />
```



## 2.137.2 ChartFormulaParameter.Source

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartFormulaParameter.Source** element specifies the name of the [ChartDataPointValues](#) element to use as the value of a [ChartFormulaParameter](#). The **ChartFormulaParameter.Source** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Following is the parent element of the **ChartFormulaParameter.Source** element.

Parent elements
<a href="#">ChartFormulaParameter</a>

The following is the XML Schema definition of the **ChartFormulaParameter.Source** element.

```
<xsd:element name="Source" type="xsd:string" minOccurs="0" />
```

## 2.137.3 ChartFormulaParameter.Value

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartFormulaParameter.Value** element specifies the value of a [ChartFormulaParameter](#). The **ChartFormulaParameter.Value** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **ChartFormulaParameter.Value** element.

Parent elements
<a href="#">ChartFormulaParameter</a>

The following is the XML Schema definition of the **ChartFormulaParameter.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="0" />
```

## 2.138 ChartSeriesCollection

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartSeriesCollection** element specifies a list of [ChartSeries](#).

The following are the parent and child elements of the **ChartSeriesCollection** element.

Parent elements
<a href="#">ChartData</a>

Child elements
<a href="#">ChartSeriesCollection.ChartSeries</a>

The following is the XML Schema definition of the **ChartSeriesCollection** element in RDL 2008/01.

```
<xsd:complexType name="ChartSeriesCollectionType">
  <xsd:sequence maxOccurs="unbounded" minOccurs="1">
    <xsd:element name="ChartSeries" type="ChartSeriesType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartSeriesCollection** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartSeriesCollectionType">
  <xsd:sequence maxOccurs="unbounded" minOccurs="1">
    <xsd:element name="ChartSeries" type="ChartSeriesType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.138.1 ChartSeriesCollection.ChartSeries

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSeriesCollection.ChartSeries** element specifies the **data points** for each series in a [Chart](#). This element MUST be specified. There MUST be as many [ChartSeries](#) elements as there are leaf-node chart members in the [Chart.ChartSeriesHierarchy](#). The **ChartSeriesCollection.ChartSeries** element is of type **ChartSeries**.

Following is the parent element of the **ChartSeriesCollection.ChartSeries** element.

Parent elements
<a href="#">ChartSeriesCollection</a>

The following is the XML Schema definition of the **ChartSeriesCollection.ChartSeries** element.

```
<xsd:element name="ChartSeries" type="ChartSeriesType" minOccurs="1"
  maxOccurs="unbounded" />
```

### 2.139 ChartDataLabel

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataLabel** element specifies the data labels to display on data values.

The following are the parent elements and child elements of the **ChartDataLabel** element.

Parent elements
<a href="#">ChartDataPoint</a>
<a href="#">ChartEmptyPoints</a>
ChartSeries

Child elements
<a href="#">ChartDataLabel.ActionInfo</a>
<a href="#">ChartDataLabel.Label</a>
<a href="#">ChartDataLabel.Position</a>
<a href="#">ChartDataLabel.Rotation</a>
<a href="#">ChartDataLabel.Style</a>
<a href="#">ChartDataLabel.ToolTip</a>
<a href="#">ChartDataLabel.UseValueAsLabel</a>
<a href="#">ChartDataLabel.Visible</a>

The following is the XML Schema definition of the **ChartDataLabel** element in RDL 2008/01.

```
<xsd:complexType name="ChartDataLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Visible" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Label" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="UseValueAsLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="Rotation" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartDataLabel** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartDataLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Visible" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Label" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="UseValueAsLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="Rotation" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.139.1 ChartDataLabel.ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataLabel.ActionInfo** element specifies action that are associated with a chart data label. This element is optional. This element is of type [ActionInfo](#).

Following is the parent element of the **ChartDataLabel.ActionInfo** element.

Parent elements
<a href="#">ChartDataLabel</a>

The following is the XML Schema definition of the **ChartDataLabel.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

## 2.139.2 ChartDataLabel.Label

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataLabel.Label** element specifies the text of a chart data label. The **ChartDataLabel.Label** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

If the value of the [ChartDataLabel.UseValueAsLabel](#) element is true, the **ChartDataLabel.Label** element is ignored.

Following is the parent element of the **ChartDataLabel.Label** element.

Parent elements
<a href="#">ChartDataLabel</a>

The following is the XML Schema definition of the **ChartDataLabel.Label** element.

```
<xsd:element name="Label" type="StringLocIDType" minOccurs="0" />
```

## 2.139.3 ChartDataLabel.Position

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataLabel.Position** element specifies the position of the chart data label for a data point. This element is optional. If the **ChartDataLabel.Position** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Auto:** Specifies automatic positioning.

**Top:** Specifies to position the label at the top of the data point.

**TopLeft:** Specifies to position the label at the top-left of the data point.

**TopRight:** Specifies to position the label at the top-right of the data point.

**Left:** Specifies to position the label to the left of the data point.

**Center:** Specifies to position the label at the center of the data point.

**Right:** Specifies to position the label at the right of the data point.

**BottomRight:** Specifies to position the label at the bottom-right of the data point.

**Bottom:** Specifies to position the label at the bottom of the data point.

**BottomLeft:** Specifies to position the label at the bottom-left of the data point.

**Outside:** Specifies to position the label outside of the data point. If the subtype of the containing series, as specified in [ChartSeries.Subtype](#), is not "Pie" or "Doughnut", "Outside" MUST be treated as "Top".

If the **ChartDataLabel.Position** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartDataLabel.Position** element.

Parent elements
<a href="#">ChartDataLabel</a>

The following is the XML Schema definition of the **ChartDataLabel.Position** element.

```
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
```

### 2.139.4 ChartDataLabel.Rotation

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDataLabel.Rotation** element specifies the angle of rotation for data label text. The **ChartDataLabel.Rotation** element is optional. If this element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 0

Following is the parent element of the **ChartDataLabel.Rotation** element.

Parent elements
<a href="#">ChartDataLabel</a>

The following is the XML Schema definition of the **ChartDataLabel.Rotation** element.

```
<xsd:element name="Rotation" type="xsd:string" minOccurs="0" />
```

### 2.139.5 ChartDataLabel.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDataLabel.Style** element specifies the style properties of a chart data label. Supplied style properties MUST override the properties of the [ChartSeries](#) styles. The **ChartDataLabel.Style** element is optional. This element is of type [Style](#).

Following is the parent element of the **ChartDataLabel.Style** element.

Parent elements
<a href="#">ChartDataLabel</a>

The following is the XML Schema definition of the **ChartDataLabel.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.139.6 ChartDataLabel.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDataLabel.ToolTip** element specifies the tooltip to display for a chart data label. The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **ChartDataLabel.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **ChartDataLabel.ToolTip**.

Parent elements
<a href="#">ChartDataLabel</a>

The following is the XML Schema definition of the **ChartDataLabel.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

## 2.139.7 ChartDataLabel.UseValueAsLabel

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDataLabel.UseValueAsLabel** element specifies whether the Y value of a data point MUST be used as a chart data label. The **ChartDataLabel.UseValueAsLabel** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartDataLabel.UseValueAsLabel** element.

Parent elements
<a href="#">ChartDataLabel</a>

The following is the XML Schema definition of the **ChartDataLabel.UseValueAsLabel** element.

```
<xsd:element name="UseValueAsLabel" type="xsd:string" minOccurs="0" />
```

## 2.139.8 ChartDataLabel.Visible

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDataLabel.Visible** element specifies whether a chart data label is displayed on a [Chart](#). The **ChartDataLabel.Visible** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartDataLabel.Visible** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartDataLabel</a>
--------------------------------

The following is the XML Schema definition of the **ChartDataLabel.Visible** element.

```
<xsd:element name="Visible" type="xsd:string" minOccurs="0" />
```

## 2.140 ChartDataPoints

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDataPoints** element specifies an ordered list of [ChartDataPoint](#) elements for a ChartSeries. There **MUST** be one corresponding **ChartDataPoints** element for each [StaticMember](#) that is specified in the [Chart.ChartCategoryHierarchy](#).

The following are the parent and child elements of the **ChartDataPoints** element.

<b>Parent elements</b>
------------------------

ChartSeries
-------------

<b>Child elements</b>
-----------------------

<a href="#">ChartDataPoints.ChartDataPoint</a>
--

The following is the XML Schema definition of the **ChartDataPoints** element in RDL 2008/01.

```
<xsd:complexType name="ChartDataPointsType" >
  <xsd:sequence>
    <xsd:element name="ChartDataPoint" type="ChartDataPointType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartDataPoints** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartDataPointsType" >
  <xsd:sequence>
    <xsd:element name="ChartDataPoint" type="ChartDataPointType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.140.1 ChartDataPoints.ChartDataPoint

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDataPoints.ChartDataPoint** element specifies a [ChartDataPoint](#) within the set of [ChartDataPoints](#) for a ChartSeries. The **ChartDataPoints.ChartDataPoint** element **MUST** be specified. This element **MUST** have at least one **ChartDataPoint** instance. This element is of type **ChartDataPoint**.

Following is the parent element of the **ChartDataPoints.ChartDataPoint** element.

Parent elements
<a href="#">ChartDataPoints</a>

The following is the XML Schema definition of the **ChartDataPoints.ChartDataPoint** element.

```
<xsd:element name="ChartDataPoint" type="ChartDataPointType" maxOccurs="unbounded" />
```

## 2.141 ChartDataPoint

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataPoint** element specifies a **data point** in a [Chart](#). This element can consist of a single value expression (for chart types such as "Bar" or "Line") or multiple value expressions (for chart types such as "Range" and "Scatter").

The following are the parent and child elements of the **ChartDataPoint** element.

Parent elements
<a href="#">ChartDataPoints</a>

Child elements
<a href="#">ChartDataPoint.ActionInfo</a>
<a href="#">ChartDataPoint.AxisLabel</a>
<a href="#">ChartDataPoint.ChartDataLabel</a>
<a href="#">ChartDataPoint.ChartDataPointValues</a>
<a href="#">ChartDataPoint.ChartItemInLegend</a>
<a href="#">ChartDataPoint.ChartMarker</a>
<a href="#">ChartDataPoint.CustomProperties</a>
<a href="#">ChartDataPoint.DataElementName</a>
<a href="#">ChartDataPoint.DataElementOutput</a>
<a href="#">ChartDataPoint.Style</a>
<a href="#">ChartDataPoint.ToolTip</a>

**Applies to** [RDL 2011/01](#)

Child elements
<a href="#">ChartDataPoint.DataSetName</a>
<a href="#">ChartDataPoint.Relationships</a>

The following is the XML Schema definition of the **ChartDataPoint** element in RDL 2008/01.



```

<xsd:complexType name="ChartDataPointType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartDataPointValues" type="ChartDataPointValuesType"
      minOccurs="0" />
    <xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
    <xsd:element name="AxisLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
    <xsd:element name="ChartItemInLegend" type="ChartItemInLegendType"
      minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **ChartDataPoint** element in RDL 2010/01 and RDL 2016/01.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```

<xsd:complexType name="ChartDataPointType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartDataPointValues" type="ChartDataPointValuesType"
      minOccurs="0" />
    <xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
    <xsd:element name="AxisLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
    <xsd:element name="ChartItemInLegend" type="ChartItemInLegendType"
      minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.141.1 ChartDataPoint.ActionInfo

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataPoint.ActionInfo** element specifies actions that are associated with a chart data point. This element is optional. This element is of type [ActionInfo](#).

Following is the parent element of the **ChartDataPoint.ActionInfo** element.

Parent elements
<a href="#">ChartDataPoint</a>

The following is the XML Schema definition of the **ChartDataPoint.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.141.2 ChartDataPoint.AxisLabel

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataPoint.AxisLabel** element specifies the label to use on an axis for a chart data point. The **ChartDataPoint.AxisLabel** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **ChartDataPoint.AxisLabel** element.

Parent elements
<a href="#">ChartDataPoint</a>

The following is the XML Schema definition of the **ChartDataPoint.AxisLabel** element.

```
<xsd:element name="AxisLabel" type="xsd:string" minOccurs="0" />
```

### 2.141.3 ChartDataPoint.ChartDataLabel

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataPoint.ChartDataLabel** element specifies a data label for a data point. This element is optional. This element is of type [ChartDataLabel](#).

Following is the parent element of the **ChartDataPoint.ChartDataLabel** element.

Parent elements
<a href="#">ChartDataPoint</a>

The following is the XML Schema definition of the **ChartDataPoint.ChartDataLabel** element.

```
<xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
```

## 2.141.4 ChartDataPoint.ChartDataPointValues

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataPoint.ChartDataPointValues** element specifies data values for a chart data point. This element is optional. This element is of type [ChartDataPointValues](#).

Following is the parent element of the **ChartDataPoint.ChartDataPointValues** element.

Parent elements
<a href="#">ChartDataPoint</a>

The following is the XML Schema definition of the **ChartDataPoint.ChartDataPointValues** element.

```
<xsd:element name="ChartDataPointValues" type="ChartDataPointValuesType"
  minOccurs="0" />
```

## 2.141.5 ChartDataPoint.ChartItemInLegend

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataPoint.ChartItemInLegend** element specifies how a data point will appear when it is displayed in a legend. This element is optional. To use this element, the value of the [ChartSeries.Type](#) element SHOULD be "Shape".

The **ChartDataPoint.ChartItemInLegend** element is of type [ChartItemInLegend](#).

Following is the parent element of the **ChartDataPoint.ChartItemInLegend** element.

Parent elements
<a href="#">ChartDataPoint</a>

The following is the XML Schema definition of the **ChartDataPoint.ChartItemInLegend** element.

```
<xsd:element name="ChartItemInLegend" type="ChartItemInLegendType"
  minOccurs="0" />
```

## 2.141.6 ChartDataPoint.ChartMarker

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataPoint.ChartMarker** element specifies the appearance of a data point marker. This element is optional. This element is of type [ChartMarker](#).

Following is the parent element of the **ChartDataPoint.ChartMarker** element.

Parent elements
<a href="#">ChartDataPoint</a>

The following is the XML Schema definition of the **ChartDataPoint.ChartMarker** element.

```
<xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
```

## 2.141.7 ChartDataPoint.CustomProperties

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataPoint.CustomProperties** element specifies custom properties for a chart data point. This element is optional. This element is of type [CustomProperties](#).

Following is the parent element of the **ChartDataPoint.CustomProperties** element.

Parent elements
<a href="#">ChartDataPoint</a>

The following is the XML Schema definition of the **ChartDataPoint.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

## 2.141.8 ChartDataPoint.DataElementName

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataPoint.DataElementName** element specifies the name to use for the **data element** or attribute for a chart data point. The **ChartDataPoint.DataElementName** element is optional. If this element is present, its value MUST be a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as the name of the corresponding **static series** or **static category**.

Following is the parent element of the **ChartDataPoint.DataElementName** element.

Parent elements
<a href="#">ChartDataPoint</a>

The following is the XML Schema definition of the **ChartDataPoint.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

## 2.141.9 ChartDataPoint.DataElementOutput

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataPoint.DataElementOutput** element specifies whether an item appears in a **data rendering**. This element is optional. If the **ChartDataPoint.DataElementOutput** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**Output:** Specifies that the data point appears in the data rendering output.

**NoOutput:** Specifies that the data point does not appear in the data rendering output.

**ContentsOnly:** Specifies that the data point does not appear in the output, but that its values appears in the data **renderer** output as if they were in the data point's container element.

If the **ChartDataPoint.DataElementOutput** element is not present, its value is interpreted as "ContentsOnly".

Following is the parent element of the **ChartDataPoint.DataElementOutput** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartDataPoint</a>
--------------------------------

The following is the XML Schema definition of the **ChartDataPoint.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.141.10 ChartDataPoint.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDataPoint.Style** element specifies the style properties for a chart data point. This element is optional. This element is of type [Style](#).

Following is the parent element of the **ChartDataPoint.Style** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartDataPoint</a>
--------------------------------

The following is the XML Schema definition of the **ChartDataPoint.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.141.11 ChartDataPoint.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDataPoint.ToolTip** element specifies the tooltip to display for a chart data point. The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **ChartDataPoint.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **ChartDataPoint.ToolTip** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartDataPoint</a>
--------------------------------

The following is the XML Schema definition of the **ChartDataPoint.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

## 2.141.12 ChartDataPoint.DataSetName

*Applies to [RDL 2011/01](#)*

The **ChartDataPoint.DataSetName** element specifies which [DataSet](#) to use for a [ChartDataPoint](#). The **ChartDataPoint.DataSetName** element is optional and MUST NOT be specified more than once.

If the **ChartDataPoint.DataSetName** element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as the **DataSetName** of the containing scope. The **ChartDataPoint.DataSetName** element MUST be specified unless all containing scopes have the same **DataSet**. The **ChartDataPoint.DataSetName** element is ignored if the **DataSet** for this **ChartDataPoint** is the same as the **DataSet** for all containing scopes. [<29><30>](#)

Following is the parent element of the **ChartDataPoint.DataSetName** element.

Parent elements
ChartDataPoint

The following is the XML Schema definition of the **ChartDataPoint.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
```

## 2.141.13 ChartDataPoint.Relationships

*Applies to [RDL 2011/01](#)*

The **ChartDataPoint.Relationships** element specifies a collection of the **relationships** to use for correlating data in this [ChartDataPoint](#) with the data in the containing scopes. The **ChartDataPoint.Relationships** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [Relationships].

This element is ignored if the [DataSet](#) for this **ChartDataPoint** is the same as the **DataSet** for each containing scope.

Following is the parent element of the **ChartDataPoint.Relationships** element.

Parent elements
ChartDataPoint

The following is the XML Schema definition of the **ChartDataPoint.Relationships** element.

```
<xsd:element name="Relationships" type="RelationshipsType" minOccurs="0" />
```

## 2.142 ChartDataPointValues

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDataPointValues** element specifies a set of data values for a chart **data point**. The set of mandatory and optional data values is determined based on the value of the type of the [ChartSeries](#).

The following are the parent and child elements of the **ChartDataPointValues** element.

<b>Parent elements</b>
<a href="#">ChartDataPoint</a>

<b>Child elements</b>
<a href="#">ChartDataPointValues.End</a>
<a href="#">ChartDataPointValues.High</a>
<a href="#">ChartDataPointValues.Low</a>
<a href="#">ChartDataPointValues.Mean</a>
<a href="#">ChartDataPointValues.Median</a>
<a href="#">ChartDataPointValues.Size</a>
<a href="#">ChartDataPointValues.Start</a>
<a href="#">ChartDataPointValues.X</a>
<a href="#">ChartDataPointValues.Y</a>

**Applies to [RDL 2011/01](#)**

<b>Child elements</b>
<a href="#">ChartDataPointValues.HighlightX</a>
<a href="#">ChartDataPointValues.HighlightY</a>
<a href="#">ChartDataPointValues.HighlightSize</a>

**Applies to [RDL 2013/01](#)**

<b>Child elements</b>
<a href="#">ChartDataPointValues.FormatX</a>
<a href="#">ChartDataPointValues.FormatY</a>
<a href="#">ChartDataPointValues.FormatSize</a>
<a href="#">ChartDataPointValues.CurrencyLanguageX</a>
<a href="#">ChartDataPointValues.CurrencyLanguageY</a>
<a href="#">ChartDataPointValues.CurrencyLanguageSize</a>

The following is the XML Schema definition of the **ChartDataPointValues** element in RDL 2008/01.

```
<xsd:complexType name="ChartDataPointValuesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="X" type="xsd:string" minOccurs="0" />
    <xsd:element name="Y" type="xsd:string" minOccurs="0" />
    <xsd:element name="Size" type="xsd:string" minOccurs="0" />
    <xsd:element name="High" type="xsd:string" minOccurs="0" />
    <xsd:element name="Low" type="xsd:string" minOccurs="0" />
    <xsd:element name="Start" type="xsd:string" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

```

    <xsd:element name="End" type="xsd:string" minOccurs="0" />
    <xsd:element name="Mean" type="xsd:string" minOccurs="0" />
    <xsd:element name="Median" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **ChartDataPointValues** element in RDL 2010/01 and RDL 2016/01.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, [5.6](#), and 5.7, respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```

<xsd:complexType name="ChartDataPointValuesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="X" type="xsd:string" minOccurs="0" />
    <xsd:element name="Y" type="xsd:string" minOccurs="0" />
    <xsd:element name="Size" type="xsd:string" minOccurs="0" />
    <xsd:element name="High" type="xsd:string" minOccurs="0" />
    <xsd:element name="Low" type="xsd:string" minOccurs="0" />
    <xsd:element name="Start" type="xsd:string" minOccurs="0" />
    <xsd:element name="End" type="xsd:string" minOccurs="0" />
    <xsd:element name="Mean" type="xsd:string" minOccurs="0" />
    <xsd:element name="Median" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.142.1 ChartDataPointValues.End

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataPointValues.End** element specifies the end or close value for a [ChartDataPoint](#). The **ChartDataPointValues.End** element is optional. This element MUST be specified for series in which the value of [ChartSeries.Type](#) is "Range" and the value of [ChartSeries.Subtype](#) is "Stock", "Candlestick", or "BoxPlot".

If the **ChartDataPointValues.End** element is present, its value MUST be a **Numeric** or an expression that evaluates to a **Numeric**.

Following is the parent element of the **ChartDataPointValues.End** element.

<b>Parent elements</b>
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.End** element.

```

<xsd:element name="End" type="xsd:string" minOccurs="0" />

```

### 2.142.2 ChartDataPointValues.High

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)



The **ChartDataPointValues.High** element specifies the high value for a [ChartDataPoint](#). This element is optional. If the value of [ChartSeries.Type](#) is "Range" and [ChartDataPointValues.Y](#) is not specified, the **ChartDataPointValues.High** element MUST be specified.

If the **ChartDataPointValues.High** element is present, its value MUST be a **Numeric** or an expression that evaluates to a **Numeric**.

Following is the parent element of the **ChartDataPointValues.High** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.High** element.

```
<xsd:element name="High" type="xsd:string" minOccurs="0" />
```

### 2.142.3 ChartDataPointValues.Low

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDataPointValues.Low** element specifies the low value for a [ChartDataPoint](#). This element is optional. If the value of [ChartSeries.Type](#) is "Range" and if the **ChartDataPointValues.Low** element is not specified, the value of the **ChartDataPointValues.Low** element is interpreted as 0.

If this element is present, its value MUST be a **Numeric** or an expression that evaluates to a **Numeric**.

Following is the parent element of the **ChartDataPointValues.Low** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.Low** element.

```
<xsd:element name="Low" type="xsd:string" minOccurs="0" />
```

### 2.142.4 ChartDataPointValues.Mean

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDataPointValues.Mean** element specifies the mean value for a [ChartDataPoint](#). This element is optional. The **ChartDataPointValues.Mean** element MUST be specified for series in which the value of [ChartSeries.Type](#) is "Range" and the value of [ChartSeries.Subtype](#) is "BoxPlot".

If the **ChartDataPointValues.Mean** element is present, its value MUST be a **Numeric** or an expression that evaluates to a **Numeric**.

Following is the parent element of the **ChartDataPointValues.Mean** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.Mean** element.

```
<xsd:element name="Mean" type="xsd:string" minOccurs="0" />
```

### 2.142.5 ChartDataPointValues.Median

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataPointValues.Median** element specifies the median value for a [ChartDataPoint](#). This element is optional. The **ChartDataPointValues.Median** element MUST be specified for series in which the value of [ChartSeries.Type](#) is "Range" and the value of [ChartSeries.Subtype](#) is "BoxPlot".

If the **ChartDataPointValues.Median** element is present, its value MUST be a **Numeric** or an expression that evaluates to a **Numeric**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **ChartDataPointValues.Median** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.Median** element.

```
<xsd:element name="Median" type="xsd:string" minOccurs="0" />
```

### 2.142.6 ChartDataPointValues.Size

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataPointValues.Size** element specifies the size value for a [ChartDataPoint](#). This element is optional. The **ChartDataPointValues.Size** element MUST be specified for series in which the value of [ChartSeries.Type](#) is "Scatter" and the value of [ChartSeries.Subtype](#) is "Bubble".

If the **ChartDataPointValues.Size** element is present, its value MUST be a **Numeric** or an expression that evaluates to a **Numeric**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **ChartDataPointValues.Size** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.Size** element.

```
<xsd:element name="Size" type="xsd:string" minOccurs="0" />
```

### 2.142.7 ChartDataPointValues.Start

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartDataPointValues.Start** element specifies the start or open value for a [ChartDataPoint](#). This element is optional. The **ChartDataPointValues.Start** element MUST be specified for series in which the value of [ChartSeries.Type](#) is "Range" and the value of [ChartSeries.Subtype](#) is "Stock", "Candlestick", or "BoxPlot".

If the **ChartDataPointValues.Start** element is present, its value MUST be a **Numeric** or an expression that evaluates to a **Numeric**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **ChartDataPointValues.Start** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.Start** element.

```
<xsd:element name="Start" type="xsd:string" minOccurs="0" />
```

## 2.142.8 ChartDataPointValues.X

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDataPointValues.X** element specifies the X value for a [ChartDataPoint](#). This element is optional. This element MUST be specified for a series if the value of [ChartSeries.Type](#) is "Scatter". If the **ChartDataPointValues.X** element is present, its value MUST be a **Numeric** or an expression that evaluates to a **Numeric**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **ChartDataPointValues.X** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.X** element.

```
<xsd:element name="X" type="xsd:string" minOccurs="0" />
```

## 2.142.9 ChartDataPointValues.Y

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartDataPointValues.Y** element specifies the Y value for a [ChartDataPoint](#). This element is optional. This element MUST be specified for a series if at least one of the following conditions is true:

- The value of [ChartSeries.Type](#) is not "Range".
- The value of **ChartSeries.Type** is "Range", and the value of [ChartSeries.Subtype](#) is "ErrorBar".

If the **ChartDataPointValues.Y** element is present, its value MUST be a **Numeric** or an expression that evaluates to a **Numeric**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **ChartDataPointValues.Y** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.Y** element.

```
<xsd:element name="Y" type="xsd:string" minOccurs="0" />
```

## 2.142.10 ChartDataPointValues.HighlightX

*Applies to [RDL 2011/01](#)*

The **ChartDataPointValues.HighlightX** element specifies the highlighted X value for a [ChartDataPoint](#). This element is optional. If the **ChartDataPointValues.HighlightX** element is present, its value MUST be a **Numeric** or an expression that evaluates to a **Numeric**. If the **ChartDataPointValues.HighlightX** element is not present, its value is interpreted as 0.

Following is the parent element of the **ChartDataPointValues.HighlightX** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.HighlightX** element.

```
<xsd:element name="HighlightX" type="xsd:string" minOccurs="0" />
```

## 2.142.11 ChartDataPointValues.HighlightY

*Applies to [RDL 2011/01](#)*

The **ChartDataPointValues.HighlightY** element specifies the highlighted Y value for a [ChartDataPoint](#). The **ChartDataPointValues.HighlightY** element is optional. If the **ChartDataPointValues.HighlightY** element is present, its value MUST be a **Numeric** or an expression that evaluates to a **Numeric**. If the **ChartDataPointValues.HighlightY** element is not present, its value is interpreted as 0.

Following is the parent element of the **ChartDataPointValues.HighlightY** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.HighlightY** element.

```
<xsd:element name="HighlightY" type="xsd:string" minOccurs="0" />
```

## 2.142.12 ChartDataPointValues.HighlightSize

*Applies to [RDL 2011/01](#)*

The **ChartDataPointValues.HighlightSize** element specifies the highlighted size value for a [ChartDataPoint](#). The **ChartDataPointValues.HighlightSize** element is optional. If the **ChartDataPointValues.HighlightSize** element is present, its value MUST be a **Numeric** or an expression that evaluates to a **Numeric**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **ChartDataPointValues.HighlightSize** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.HighlightSize** element.

```
<xsd:element name="HighlightSize" type="xsd:string" minOccurs="0" />
```

### 2.142.13 ChartDataPointValues.FormatX

**Applies to** [RDL 2013/01](#)

The **ChartDataPointValues.FormatX** element specifies a formatting string that is used for the [ChartDataPointValues.X](#) element. The **ChartDataPointValues.FormatX** element is optional.

If the **ChartDataPointValues.FormatX** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, no formatting exists for the associated text to which the **ChartDataPointValues.FormatX** element applies.

If the **ChartDataPointValues.FormatX** element defines a locale-dependent currency format string, it uses [ChartDataPointValues.CurrencyLanguageX](#) as the locale.

Following is the parent element of the **ChartDataPointValues.FormatX** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.FormatX** element.

```
<xsd:element name="FormatX" type="xsd:string" />
```

### 2.142.14 ChartDataPointValues.FormatY

**Applies to** [RDL 2013/01](#)

The **ChartDataPointValues.FormatY** element specifies a formatting string that is used for the [ChartDataPointValues.Y](#) element. The **ChartDataPointValues.FormatY** element is optional.

If the **ChartDataPointValues.FormatY** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, no formatting exists for the associated text to which the **ChartDataPointValues.FormatY** element applies.

If the **ChartDataPointValues.FormatY** element defines a locale-dependent currency format string, it uses [ChartDataPointValues.CurrencyLanguageY](#) as the locale.

Following is the parent element of the **ChartDataPointValues.FormatY** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.FormatY** element.

```
<xsd:element name="FormatY" type="xsd:string" />
```

## 2.142.15 ChartDataPointValues.FormatSize

**Applies to** [RDL 2013/01](#)

The **ChartDataPointValues.FormatSize** element specifies a formatting string that is used for the [ChartDataPointValues.Size](#) element. The **ChartDataPointValues.FormatSize** element is optional.

If the **ChartDataPointValues.FormatSize** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, no formatting exists for the associated text to which the **ChartDataPointValues.FormatSize** element applies.

If the **ChartDataPointValues.FormatSize** element defines a locale-dependent currency format string, it uses [ChartDataPointValues.CurrencyLanguageSize](#) as the locale.

Following is the parent element of the **ChartDataPointValues.FormatSize** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.FormatSize** element.

```
<xsd:element name="FormatSize" type="xsd:string" />
```

## 2.142.16 ChartDataPointValues.CurrencyLanguageX

**Applies to** [RDL 2013/01](#)

The **ChartDataPointValues.CurrencyLanguageX** element specifies the currency language that is used for the [ChartDataPointValues.X](#) element. The **ChartDataPointValues.CurrencyLanguageX** element is optional.

If the **ChartDataPointValues.CurrencyLanguageX** element is present, its value MUST be a [ReportLanguage](#) value. If the **ChartDataPointValues.CurrencyLanguageX** element is not present, its value MUST be the same as that of the [Report.Language](#) element.

Following is the parent element of the **ChartDataPointValues.CurrencyLanguageX** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.CurrencyLanguageX** element.

```
<xsd:element name="CurrencyLanguageX" type="xsd:string" />
```

## 2.142.17 ChartDataPointValues.CurrencyLanguageY

**Applies to** [RDL 2013/01](#)

The **ChartDataPointValues.CurrencyLanguageY** element specifies the currency language that is used for the [ChartDataPointValues.Y](#) element. The **ChartDataPointValues.CurrencyLanguageY** element is optional.

If the **ChartDataPointValues.CurrencyLanguageY** element is present, its value MUST be a [ReportLanguage](#) value. If the **ChartDataPointValues.CurrencyLanguageY** element is not present, its value MUST be the same as that of the [Report.Language](#) element.

Following is the parent element of the **ChartDataPointValues.CurrencyLanguageY** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.CurrencyLanguageY** element.

```
<xsd:element name="CurrencyLanguageY" type="xsd:string" />
```

## 2.142.18 ChartDataPointValues.CurrencyLanguageSize

*Applies to [RDL 2013/01](#)*

The **ChartDataPointValues.CurrencyLanguageSize** element specifies the currency language that is used for the [ChartDataPointValues.Size](#) element. The **ChartDataPointValues.CurrencyLanguageSize** element is optional.

If the **ChartDataPointValues.CurrencyLanguageSize** element is present, its value MUST be a [ReportLanguage](#) value. If the **ChartDataPointValues.CurrencyLanguageSize** element is not present, its value MUST be the same as that of the [Report.Language](#) element.

Following is the parent element of the **ChartDataPointValues.CurrencyLanguageSize** element.

Parent elements
<a href="#">ChartDataPointValues</a>

The following is the XML Schema definition of the **ChartDataPointValues.CurrencyLanguageSize** element.

```
<xsd:element name="CurrencyLanguageSize" type="xsd:string" />
```

## 2.143 ChartEmptyPoints

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartEmptyPoints** element specifies the behavior for an empty data point in a [ChartSeries](#).

The following are the parent and child elements of the **ChartEmptyPoints** element.

Parent elements
ChartSeries

Child elements
<a href="#">ChartEmptyPoints.ActionInfo</a>

Child elements
<a href="#">ChartEmptyPoints.AxisLabel</a>
<a href="#">ChartEmptyPoints.ChartDataLabel</a>
<a href="#">ChartEmptyPoints.ChartMarker</a>
<a href="#">ChartEmptyPoints.CustomProperties</a>
<a href="#">ChartEmptyPoints.Style</a>
<a href="#">ChartEmptyPoints.ToolTip</a>

The following is the XML Schema definition of the **ChartEmptyPoints** element in RDL 2008/01.

```
<xsd:complexType name="ChartEmptyPointsType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
    <xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
    <xsd:element name="AxisLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartEmptyPoints** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartEmptyPointsType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
    <xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
    <xsd:element name="AxisLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.143.1 ChartEmptyPoints.ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartEmptyPoints.ActionInfo** element specifies actions that are associated with an empty data point. The **ChartEmptyPoints.ActionInfo** element is optional. This element is of type [ActionInfo](#).

Following is the parent element of the **ChartEmptyPoints.ActionInfo** element.

Parent elements
<a href="#">ChartEmptyPoints</a>



The following is the XML Schema definition of the **ChartEmptyPoints.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.143.2 ChartEmptyPoints.AxisLabel

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartEmptyPoints.AxisLabel** element specifies the label to use on the axis of an empty data point. The **ChartEmptyPoints.AxisLabel** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **ChartEmptyPoints.AxisLabel** element.

<b>Parent elements</b>
<a href="#">ChartEmptyPoints</a>

The following is the XML Schema definition of the **ChartEmptyPoints.AxisLabel** element.

```
<xsd:element name="AxisLabel" type="xsd:string" minOccurs="0" />
```

### 2.143.3 ChartEmptyPoints.ChartDataLabel

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartEmptyPoints.ChartDataLabel** element specifies a [ChartDataLabel](#) for an empty data point. The **ChartEmptyPoints.ChartDataLabel** element is optional. This element is of type **ChartDataLabel**.

Following is the parent element of the **ChartEmptyPoints.ChartDataLabel** element.

<b>Parent elements</b>
<a href="#">ChartEmptyPoints</a>

The following is the XML Schema definition of the **ChartEmptyPoints.ChartDataLabel** element.

```
<xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
```

### 2.143.4 ChartEmptyPoints.ChartMarker

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartEmptyPoints.ChartMarker** element specifies the appearance of the mark for an empty data point. The **ChartEmptyPoints.ChartMarker** element is optional. This element is of type [ChartMarker](#).

Following is the parent element of the **ChartEmptyPoints.ChartMarker** element.

<b>Parent elements</b>
<a href="#">ChartEmptyPoints</a>

The following is the XML Schema definition of the **ChartEmptyPoints.ChartMarker** element.

```
<xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
```

### 2.143.5 ChartEmptyPoints.CustomProperties

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartEmptyPoints.CustomProperties** element specifies the custom properties for an empty data point. The **ChartEmptyPoints.CustomProperties** element is optional. This element is of type [CustomProperties](#).

Following is the parent element of the **ChartEmptyPoints.CustomProperties** element.

Parent elements
<a href="#">ChartEmptyPoints</a>

The following is the XML Schema definition of the **ChartEmptyPoints.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

### 2.143.6 ChartEmptyPoints.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartEmptyPoints.Style** element specifies the style properties for an empty [ChartDataPoint](#). The **ChartEmptyPoints.Style** element is optional. This element is of type [Style](#).

Following is the parent element of the **ChartEmptyPoints.Style** element.

Parent elements
<a href="#">ChartEmptyPoints</a>

The following is the XML Schema definition of the **ChartEmptyPoints.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.143.7 ChartEmptyPoints.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartEmptyPoints.ToolTip** element specifies the tooltip to display for an empty data point. The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **ChartEmptyPoints.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **ChartEmptyPoints.ToolTip** element.

Parent elements
<a href="#">ChartEmptyPoints</a>

The following is the XML Schema definition of the **ChartEmptyPoints.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

## 2.144 ChartItemInLegend

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartItemInLegend** element specifies the behavior for a [ChartDataPoint](#) that is displayed in a legend.

The following are the parent elements and child elements of the **ChartItemInLegend** element.

Parent elements
<a href="#">ChartDataPoint</a>
<a href="#">ChartSeries</a>

Child elements
<a href="#">ChartItemInLegend.ActionInfo</a>
<a href="#">ChartItemInLegend.Hidden</a>
<a href="#">ChartItemInLegend.LegendText</a>
<a href="#">ChartItemInLegend.ToolTip</a>

The following is the XML Schema definition of the **ChartItemInLegend** element in RDL 2008/01.

```
<xsd:complexType name="ChartItemInLegendType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartItemInLegend** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartItemInLegendType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.144.1 ChartItemInLegend.ActionInfo

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartItemInLegend.ActionInfo** element specifies actions that are associated with items in a legend. This element is optional. This element is of type [ActionInfo](#).

Following is the parent element of the **ChartItemInLegend.ActionInfo** element.

Parent elements
<a href="#">ChartItemInLegend</a>

The following is the XML Schema definition of the **ChartItemInLegend.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.144.2 ChartItemInLegend.Hidden

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartItemInLegend.Hidden** element specifies whether an item is not shown in the legend. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartItemInLegend.Hidden** element.

Parent elements
<a href="#">ChartItemInLegend</a>

The following is the XML Schema definition of the **ChartItemInLegend.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

### 2.144.3 ChartItemInLegend.LegendText

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartItemInLegend.LegendText** element specifies a label to use in a legend for a **chart item**. This element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

If the **ChartItemInLegend.LegendText** element is not present and if the parent is a [ChartDataPoint](#) element, [ChartMember.Label](#) elements will be used from the [Chart.ChartCategoryHierarchy](#) element, concatenated with a hyphen character (-) between each member.

If the **ChartItemInLegend.LegendText** element is not present and if the parent of the chart item is not a **ChartDataPoint** element, its value is interpreted as an empty string.

Following is the parent element of the **ChartItemInLegend.LegendText** element.

Parent elements
<a href="#">ChartItemInLegend</a>

The following is the XML Schema definition of the **ChartItemInLegend.LegendText** element.

```
<xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
```

#### 2.144.4 ChartItemInLegend.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartItemInLegend.ToolTip** element specifies the tooltip to display for an item in a legend. The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **ChartItemInLegend.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **ChartItemInLegend.ToolTip** element.

Parent elements
<a href="#">ChartItemInLegend</a>

The following is the XML Schema definition of the **ChartItemInLegend.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

#### 2.145 ChartMarker

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartMarker** element specifies a marker for displayed chart values.

The following are the parent elements and child elements of the **ChartMarker** element.

Parent elements
<a href="#">ChartDataPoint</a>
<a href="#">ChartEmptyPoints</a>
<a href="#">ChartLegendCustomItem</a>
<a href="#">ChartSeries</a>

Child elements
<a href="#">ChartMarker.Size</a>
<a href="#">ChartMarker.Style</a>
<a href="#">ChartMarker.Type</a>

The following is the XML Schema definition of the **ChartMarker** element in RDL 2008/01.

```
<xsd:complexType name="ChartMarkerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <xsd:element name="Size" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartMarker** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartMarkerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <xsd:element name="Size" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.145.1 ChartMarker.Size

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartMarker.Size** element specifies the height and width of the plotting area of markers. This element is optional. The value of this element MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**. If this element is not present, its value is interpreted as 3.75pt.

Following is the parent element of the **ChartMarker.Size** element.

Parent elements
<a href="#">ChartMarker</a>

The following is the XML Schema definition of the **ChartMarker.Size** element.

```
<xsd:element name="Size" type="xsd:string" minOccurs="0" />
```

### 2.145.2 ChartMarker.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartMarker.Style** element specifies the style properties for a marker. This element is optional. This element is of type [Style](#).

Following is the parent element of the **ChartMarker.Style** element.

Parent elements
<a href="#">ChartMarker</a>

The following is the XML Schema definition of the **ChartMarker.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.145.3 ChartMarker.Type

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartMarker.Type** element specifies the type of a marker. This element is optional.

If the **ChartMarker.Type** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**None**: Specifies that the marker is not displayed.

**Square**: Specifies a square marker.

**Circle**: Specifies a circle marker.

**Diamond**: Specifies a diamond marker.

**Triangle**: Specifies a triangle marker.

**Cross**: Specifies a cross marker.

**Star4**: Specifies a four-pointed star marker.

**Star5**: Specifies a five-pointed star marker.

**Star6**: Specifies a six-pointed star marker.

**Star10**: Specifies a ten-pointed star marker.

**Auto**: Specifies to automatically cycle through marker types for each series.

If the **ChartMarker.Type** element is not present, its value is interpreted as "None".

Following is the parent element of the **ChartMarker.Type** element.

Parent elements
<a href="#">ChartMarker</a>

The following is the XML Schema definition of the **ChartMarker.Type** element.

```
<xsd:element name="Type" type="xsd:string" minOccurs="0" />
```

### 2.146 ChartSmartLabel

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSmartLabel** element specifies the behavior of smart labels for a ChartSeries. This element MUST be specified.

The following are the parent and child elements of the **ChartSmartLabel** element.

<b>Parent elements</b>
ChartSeries

<b>Child elements</b>
<a href="#">ChartSmartLabel.AllowOutsidePlotArea</a>
<a href="#">ChartSmartLabel.CalloutBackColor</a>
<a href="#">ChartSmartLabel.CalloutLineAnchor</a>
<a href="#">ChartSmartLabel.CalloutLineColor</a>
<a href="#">ChartSmartLabel.CalloutLineStyle</a>
<a href="#">ChartSmartLabel.CalloutLineWidth</a>
<a href="#">ChartSmartLabel.CalloutStyle</a>
<a href="#">ChartSmartLabel.ChartNoMoveDirections</a>
<a href="#">ChartSmartLabel.Disabled</a>
<a href="#">ChartSmartLabel.MarkerOverlapping</a>
<a href="#">ChartSmartLabel.MaxMovingDistance</a>
<a href="#">ChartSmartLabel.MinMovingDistance</a>
<a href="#">ChartSmartLabel.ShowOverlapped</a>

The following is the XML Schema definition of the **ChartSmartLabel** element in RDL 2008/01.

```
<xsd:complexType name="ChartSmartLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Disabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowOutsidePlotArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutBackColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineAnchor" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowOverlapped" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerOverlapping" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaxMovingDistance" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinMovingDistance" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartNoMoveDirections" type="ChartNoMoveDirectionsType"
      minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartSmartLabel** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartSmartLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
```



```

<xsd:element name="Disabled" type="xsd:string" minOccurs="0" />
<xsd:element name="AllowOutsidePlotArea" type="xsd:string" minOccurs="0" />
<xsd:element name="CalloutBackColor" type="xsd:string" minOccurs="0" />
<xsd:element name="CalloutLineAnchor" type="xsd:string" minOccurs="0" />
<xsd:element name="CalloutLineColor" type="xsd:string" minOccurs="0" />
<xsd:element name="CalloutLineStyle" type="xsd:string" minOccurs="0" />
<xsd:element name="CalloutLineWidth" type="xsd:string" minOccurs="0" />
<xsd:element name="CalloutStyle" type="xsd:string" minOccurs="0" />
<xsd:element name="ShowOverlapped" type="xsd:string" minOccurs="0" />
<xsd:element name="MarkerOverlapping" type="xsd:string" minOccurs="0" />
<xsd:element name="MaxMovingDistance" type="xsd:string" minOccurs="0" />
<xsd:element name="MinMovingDistance" type="xsd:string" minOccurs="0" />
<xsd:element name="ChartNoMoveDirections" type="ChartNoMoveDirectionsType"
minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.146.1 ChartSmartLabel.AllowOutsidePlotArea

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSmartLabel.AllowOutsidePlotArea** element specifies whether data point labels can be drawn outside of a **plot area**. This element is optional. If the **ChartSmartLabel.AllowOutsidePlotArea** element is present, its value **MUST** be a [String](#) ([[XMLSCHEMA2/2](#)] section 3.2.1) or an expression that evaluates to a **String**.

The value of this element **MUST** be one of the following or an expression that evaluates to one of the following:

**Partial:** Specifies that labels are allowed to be partially outside the plot area.

**True:** Specifies that labels are allowed to be entirely outside the plot area.

**False:** Specifies that labels are required to be entirely inside the plot area.

If the **ChartSmartLabel.AllowOutsidePlotArea** element is not present, its value is interpreted as "Partial".

Following is the parent element of the **ChartSmartLabel.AllowOutsidePlotArea** element.

Parent elements
<a href="#">ChartSmartLabel</a>

The following is the XML Schema definition of the **ChartSmartLabel.AllowOutsidePlotArea** element.

```

<xsd:element name="AllowOutsidePlotArea" type="xsd:string" minOccurs="0" />

```

### 2.146.2 ChartSmartLabel.CalloutBackColor

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSmartLabel.CalloutBackColor** element specifies the fill color of the box that surrounds the point label text when the value of the [ChartSmartLabel.CalloutStyle](#) element is "Box".

The **ChartSmartLabel.CalloutBackColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. If this element is not present, its value is interpreted as an empty color.

Following is the parent element of the **ChartSmartLabel.CalloutBackColor** element.

Parent elements
<a href="#">ChartSmartLabel</a>

The following is the XML Schema definition of the **ChartSmartLabel.CalloutBackColor** element.

```
<xsd:element name="CalloutBackColor" type="xsd:string" minOccurs="0" />
```

### 2.146.3 ChartSmartLabel.CalloutLineAnchor

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSmartLabel.CalloutLineAnchor** element specifies the shape to be drawn on the point end of a callout line. This element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**None**: Specifies that the shape does not appear on the point end.

**Arrow**: Specifies an arrow shape.

**Diamond**: Specifies a diamond shape.

**Square**: Specifies a square shape.

**Round**: Specifies a round shape.

If the **ChartSmartLabel.CalloutLineAnchor** element is not present, its value is interpreted as "Arrow".

Following is the parent element of the **ChartSmartLabel.CalloutLineAnchor** element.

Parent elements
<a href="#">ChartSmartLabel</a>

The following is the XML Schema definition of the **ChartSmartLabel.CalloutLineAnchor** element.

```
<xsd:element name="CalloutLineAnchor" type="xsd:string" minOccurs="0" />
```

### 2.146.4 ChartSmartLabel.CalloutLineColor

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSmartLabel.CalloutLineColor** element specifies the color of a callout line. This element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. If this element is not present, its value is interpreted as "Black".

Following is the parent element of the **ChartSmartLabel.CalloutLineColor** element.

Parent elements
<a href="#">ChartSmartLabel</a>

The following is the XML Schema definition of the **ChartSmartLabel.CalloutLineColor** element.

```
<xsd:element name="CalloutLineColor" type="xsd:string" minOccurs="0" />
```

## 2.146.5 ChartSmartLabel.CalloutLineStyle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartSmartLabel.CalloutLineStyle** element specifies the style of a callout line. This element is optional.

If the **ChartSmartLabel.CalloutLineStyle** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Solid**: Specifies a solid line.

**None**: Specifies no line.

**Dotted**: Specifies a dotted line.

**Dashed**: Specifies a dashed line.

**Double**: Specifies a double solid line.

**DashDot**: Specifies a dash-dot line.

**DashDotDot**: Specifies a dash-dot-dot line.

If the **ChartSmartLabel.CalloutLineStyle** element is not present, its value is interpreted as "Solid".

Following is the parent element of the **ChartSmartLabel.CalloutLineStyle** element.

Parent elements
<a href="#">ChartSmartLabel</a>

The following is the XML Schema definition of the **ChartSmartLabel.CalloutLineStyle** element.

```
<xsd:element name="CalloutLineStyle" type="xsd:string" minOccurs="0" />
```

## 2.146.6 ChartSmartLabel.CalloutLineWidth

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartSmartLabel.CalloutLineWidth** element specifies the width of a callout line. This element is optional.

If the **ChartSmartLabel.CalloutLineWidth** element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**. If this element is not present, its value is interpreted as 0.75pt. [<31>](#)

Following is the parent element of the **ChartSmartLabel.CalloutLineWidth** element.

Parent elements
<a href="#">ChartSmartLabel</a>

The following is the XML Schema definition of the **ChartSmartLabel.CalloutLineWidth** element.

```
<xsd:element name="CalloutLineWidth" type="xsd:string" minOccurs="0" />
```

### 2.146.7 ChartSmartLabel.CalloutStyle

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSmartLabel.CalloutStyle** element specifies the style to use when drawing callout lines. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Underline:** Specifies to attach the callout line to an underline on the label.

**Box:** Specifies to attach the callout line to a box around the label.

**None:** Specifies no additional label style for the callout line.

If the **ChartSmartLabel.CalloutStyle** element is not present, its value is interpreted as "Underline".

Following is the parent element of the **ChartSmartLabel.CalloutStyle** element.

Parent elements
<a href="#">ChartSmartLabel</a>

The following is the XML Schema definition of the **ChartSmartLabel.CalloutStyle** element.

```
<xsd:element name="CalloutStyle" type="xsd:string" minOccurs="0" />
```

### 2.146.8 ChartSmartLabel.ChartNoMoveDirections

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSmartLabel.ChartNoMoveDirection** element specifies the directions in which a label is not allowed to move. This element is optional and is of type [ChartNoMoveDirections](#).

Following is the parent element of the **ChartSmartLabel.ChartNoMoveDirection** element.

Parent elements
<a href="#">ChartSmartLabel</a>

The following is the XML Schema definition of the **ChartSmartLabel.ChartNoMoveDirection** element.

```
<xsd:element name="ChartNoMoveDirections" type="ChartNoMoveDirectionsType"
```

```
minOccurs="0" />
```

### 2.146.9 ChartSmartLabel.Disabled

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSmartLabel.Disabled** element specifies whether smart labels are turned off. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartSmartLabel.Disabled** element.

Parent elements
<a href="#">ChartSmartLabel</a>

The following is the XML Schema definition of the **ChartSmartLabel.Disabled** element.

```
<xsd:element name="Disabled" type="xsd:string" minOccurs="0" />
```

### 2.146.10 ChartSmartLabel.MarkerOverlapping

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSmartLabel.MarkerOverlapping** element indicates whether point labels are allowed to overlap point markers. The **ChartSmartLabel.MarkerOverlapping** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartSmartLabel.MarkerOverlapping** element.

Parent elements
<a href="#">ChartSmartLabel</a>

The following is the XML Schema definition of the **ChartSmartLabel.MarkerOverlapping** element.

```
<xsd:element name="MarkerOverlapping" type="xsd:string" minOccurs="0" />
```

### 2.146.11 ChartSmartLabel.MaxMovingDistance

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSmartLabel.MaxMovingDistance** element specifies the maximum distance from the data point that data point labels are allowed to be moved in case of overlapping. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**. If this element is not present, its value is interpreted as 23pt.

Following is the parent element of the **ChartSmartLabel.MaxMovingDistance** element.

Parent elements
<a href="#">ChartSmartLabel</a>

The following is the XML Schema definition of the **ChartSmartLabel.MaxMovingDistance** element.

```
<xsd:element name="MaxMovingDistance" type="xsd:string" minOccurs="0" />
```

## 2.146.12 ChartSmartLabel.MinMovingDistance

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSmartLabel.MinMovingDistance** element specifies the minimum distance from the data point that data point labels MUST be moved in case of overlapping. This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**. If this element is not present, its value is interpreted as Opt.

Following is the parent element of the **ChartSmartLabel.MinMovingDistance** element.

Parent elements
<a href="#">ChartSmartLabel</a>

The following is the XML Schema definition of the **ChartSmartLabel.MinMovingDistance** element.

```
<xsd:element name="MinMovingDistance" type="xsd:string" minOccurs="0" />
```

## 2.146.13 ChartSmartLabel.ShowOverlapped

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartSmartLabel.ShowOverlapped** element specifies whether labels are displayed even when overlapping issues cannot be resolved. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartSmartLabel.ShowOverlapped** element.

Parent elements
<a href="#">ChartSmartLabel</a>

The following is the XML Schema definition of the **ChartSmartLabel.ShowOverlapped** element.

```
<xsd:element name="ShowOverlapped" type="xsd:string" minOccurs="0" />
```

## 2.147 ChartNoMoveDirections

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartNoMoveDirections** element specifies the directions in which a smart label is not allowed to move.

The following are the parent and child elements of the **ChartNoMoveDirections** element.

**Parent elements**[ChartSmartLabel](#)**Child elements**[ChartNoMoveDirections.Down](#)[ChartNoMoveDirections.DownLeft](#)[ChartNoMoveDirections.DownRight](#)[ChartNoMoveDirections.Left](#)[ChartNoMoveDirections.Right](#)[ChartNoMoveDirections.Up](#)[ChartNoMoveDirections.UpLeft](#)[ChartNoMoveDirections.UpRight](#)

The following is the XML Schema definition of the **ChartNoMoveDirections** element in RDL 2008/01.

```
<xsd:complexType name="ChartNoMoveDirectionsType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Up" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Right" type="xsd:string" minOccurs="0" />
    <xsd:element name="Down" type="xsd:string" minOccurs="0" />
    <xsd:element name="UpLeft" type="xsd:string" minOccurs="0" />
    <xsd:element name="UpRight" type="xsd:string" minOccurs="0" />
    <xsd:element name="DownLeft" type="xsd:string" minOccurs="0" />
    <xsd:element name="DownRight" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartNoMoveDirections** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartNoMoveDirectionsType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Up" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Right" type="xsd:string" minOccurs="0" />
    <xsd:element name="Down" type="xsd:string" minOccurs="0" />
    <xsd:element name="UpLeft" type="xsd:string" minOccurs="0" />
    <xsd:element name="UpRight" type="xsd:string" minOccurs="0" />
    <xsd:element name="DownLeft" type="xsd:string" minOccurs="0" />
    <xsd:element name="DownRight" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.147.1 ChartNoMoveDirections.Down

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartNoMoveDirections.Down** element specifies whether a [ChartSmartLabel](#) cannot move straight down. The **ChartNoMoveDirections.Down** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartNoMoveDirections.Down** element.

Parent elements
<a href="#">ChartNoMoveDirections</a>

The following is the XML Schema definition of the **ChartNoMoveDirections.Down** element.

```
<xsd:element name="Down" type="xsd:string" minOccurs="0" />
```

### 2.147.2 ChartNoMoveDirections.DownLeft

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartNoMoveDirections.DownLeft** element specifies whether a [ChartSmartLabel](#) cannot move down and left. The **ChartNoMoveDirections.DownLeft** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartNoMoveDirections.DownLeft** element.

Parent elements
<a href="#">ChartNoMoveDirections</a>

The following is the XML Schema definition of the **ChartNoMoveDirections.DownLeft** element.

```
<xsd:element name="DownLeft" type="xsd:string" minOccurs="0" />
```

### 2.147.3 ChartNoMoveDirections.DownRight

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartNoMoveDirections.DownRight** element specifies whether a [ChartSmartLabel](#) cannot move down and right. The **ChartNoMoveDirections.DownRight** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartNoMoveDirections.DownRight** element.

Parent elements
<a href="#">ChartNoMoveDirections</a>

The following is the XML Schema definition of the **ChartNoMoveDirections.DownRight** element.



```
<xsd:element name="DownRight" type="xsd:string" minOccurs="0" />
```

#### 2.147.4 **ChartNoMoveDirections.Left**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartNoMoveDirections.Left** element specifies whether a [ChartSmartLabel](#) cannot move to the left. The **ChartNoMoveDirections.Left** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartNoMoveDirections.Left** element.

Parent elements
<a href="#">ChartNoMoveDirections</a>

The following is the XML Schema definition of the **ChartNoMoveDirections.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

#### 2.147.5 **ChartNoMoveDirections.Right**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartNoMoveDirections.Right** element specifies whether a [ChartSmartLabel](#) cannot move to the right. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartNoMoveDirections.Right** element.

Parent elements
<a href="#">ChartNoMoveDirections</a>

The following is the XML Schema definition of the **ChartNoMoveDirections.Right** element.

```
<xsd:element name="Right" type="xsd:string" minOccurs="0" />
```

#### 2.147.6 **ChartNoMoveDirections.Up**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartNoMoveDirections.Up** element specifies whether a [ChartSmartLabel](#) cannot move straight up. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartNoMoveDirections.Up** element.

Parent elements
<a href="#">ChartNoMoveDirections</a>

The following is the XML Schema definition of the **ChartNoMoveDirections.Up** element.

```
<xsd:element name="Up" type="xsd:string" minOccurs="0" />
```

### 2.147.7 ChartNoMoveDirections.UpLeft

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartNoMoveDirections.UpLeft** element specifies whether a [ChartSmartLabel](#) cannot move up and left. The **ChartNoMoveDirections.UpLeft** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartNoMoveDirections.UpLeft** element.

Parent elements
<a href="#">ChartNoMoveDirections</a>

The following is the XML Schema definition of the **ChartNoMoveDirections.UpLeft** element.

```
<xsd:element name="UpLeft" type="xsd:string" minOccurs="0" />
```

### 2.147.8 ChartNoMoveDirections.UpRight

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartNoMoveDirections.UpRight** element specifies whether a [ChartSmartLabel](#) cannot move up and right. The **ChartNoMoveDirections.UpRight** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartNoMoveDirections.UpRight** element.

Parent elements
<a href="#">ChartNoMoveDirections</a>

The following is the XML Schema definition of the **ChartNoMoveDirections.UpRight** element.

```
<xsd:element name="UpRight" type="xsd:string" minOccurs="0" />
```

## 2.148 ChartHierarchy

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartHierarchy** element specifies the hierarchy of category or series members for a [Chart](#).

The following are the parent and child element of the **ChartHierarchy** element.

Parent elements
Chart

<b>Child elements</b>
-----------------------

<a href="#">ChartHierarchy.ChartMembers</a>
---

**Applies to** [RDL 2012/01](#)

The following is an additional child element of the **ChartHierarchy** element.

<b>Child elements</b>
-----------------------

<a href="#">ChartHierarchy.EnableDrilldown</a>
--

The following is the XML Schema definition of the **ChartHierarchy** element in RDL 2008/01.

```
<xsd:complexType name="ChartHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ChartMembers" type="ChartMembersType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartHierarchy** element in RDL 2010/01 and RDL 2016/01.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections [5.5](#), [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:complexType name="ChartHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ChartMembers" type="ChartMembersType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.148.1 **ChartHierarchy.ChartMembers**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartHierarchy.ChartMembers** element specifies the collection of [ChartMember](#) elements at the base of a hierarchy. The **ChartHierarchy.ChartMembers** element **MUST** be specified. This element is of type [ChartMembers](#).

Following is the parent element of the **ChartHierarchy.ChartMembers** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartHierarchy</a>
--------------------------------

The following is the XML Schema definition of the **ChartHierarchy.ChartMembers** element.

```
<xsd:element name="ChartMembers" type="ChartMembersType" minOccurs="1"
maxOccurs="1" />
```

## 2.148.2 ChartHierarchy.EnableDrilldown

**Applies to** [RDL 2012/01](#)

The **ChartHierarchy.EnableDrilldown** element specifies whether drilldown is enabled for the member hierarchy. If **ChartHierarchy.EnableDrilldown** is specified, its value MUST be [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). A value of true specifies that the drilldown interactivity is enabled and that a portion of the hierarchy appears. The drilldown filter state determines which portion of the hierarchy appears. If the **ChartHierarchy.EnableDrilldown** element is not specified, its value is interpreted as false.

Following is the parent element of the **ChartHierarchy.EnableDrilldown** element.

Parent elements
<a href="#">ChartHierarchy</a>

The following is the XML Schema definition of the **ChartHierarchy.EnableDrilldown** element.

```
<xsd:element name="EnableDrilldown" type="xsd:boolean" />
```

## 2.149 ChartMembers

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartMembers** element specifies an ordered list of [ChartMember](#) elements for a [ChartHierarchy](#) or for submembers of a **ChartMember**.

The following are the parent and child elements of the **ChartMembers** element.

Parent elements
<a href="#">ChartHierarchy</a>
<a href="#">ChartMember</a>

Child elements
<a href="#">ChartMembers.ChartMember</a>

The following is the XML Schema definition of the **ChartMembers** element in RDL 2008/01.

```
<xsd:complexType name="ChartMembersType">
  <xsd:sequence minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ChartMember" type="ChartMemberType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartMembers** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartMembersType">
  <xsd:sequence minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ChartMember" type="ChartMemberType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.149.1 ChartMembers.ChartMember

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartMembers.ChartMember** element specifies a [ChartMember](#) element for a [ChartMembers](#) element. At least one instance of the **ChartMembers.ChartMember** element MUST be specified in the containing **ChartMembers** element. The **ChartMembers.ChartMember** element is of type **ChartMember**.

Following is the parent element of the **ChartMembers.ChartMember** element.

Parent elements
<a href="#">ChartMembers</a>

The following is the XML Schema definition of the **ChartMembers.ChartMember** element.

```
<xsd:element name="ChartMember" type="ChartMemberType" minOccurs="1"
  maxOccurs="unbounded" />
```

### 2.150 ChartMember

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartMember** element specifies a category or series **member** for a [Chart](#).

The following are the parent elements and child elements of the **ChartMember** element.

Parent elements
<a href="#">ChartMembers</a>

Child elements
<a href="#">ChartMember.ChartMembers</a>
<a href="#">ChartMember.CustomProperties</a>
<a href="#">ChartMember.DataElementName</a>
<a href="#">ChartMember.DataElementOutput</a>
<a href="#">ChartMember.Group</a>

Child elements
<a href="#">ChartMember.Label</a>
<a href="#">ChartMember.SortExpressions</a>

The following is the XML Schema definition of the **ChartMember** element in RDL 2008/01.

```
<xsd:complexType name="ChartMemberType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="ChartMembers" type="ChartMembersType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="Label" type="StringLocIDType" minOccurs="1"
      maxOccurs="1" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0" maxOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartMember** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartMemberType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="ChartMembers" type="ChartMembersType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="Label" type="StringLocIDType" minOccurs="1"
      maxOccurs="1" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0" maxOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.150.1 ChartMember.ChartMembers

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartMember.ChartMembers** element specifies the set of submembers that are contained in a chart member. This element is optional. This element is of type [ChartMembers](#).

Following is the parent element of the **ChartMember.ChartMembers** element.

Parent elements
<a href="#">ChartMember</a>

The following is the XML Schema definition of the **ChartMember.ChartMembers** element.

```
<xsd:element name="ChartMembers" type="ChartMembersType" minOccurs="0"
maxOccurs="1" />
```

### 2.150.2 ChartMember.CustomProperties

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartMember.CustomProperties** element specifies the custom properties for a chart member. This element is optional. This element is of type [CustomProperties](#).

Following is the parent element of the **ChartMember.CustomProperties** element.

Parent elements
<a href="#">ChartMember</a>

The following is the XML Schema definition of the **ChartMember.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0"
maxOccurs="1" />
```

### 2.150.3 ChartMember.DataElementName

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartMember.DataElementName** element specifies the name to use for the **data element** for a chart member. This element is optional.

If the **ChartMember.DataElementName** element is present, its value MUST be a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, the value of this element is interpreted as either the value of the [Group.Name](#) attribute for this member (if the [ChartMember.Group](#) element is present) or the value of the [ChartMember.Label](#) element.

Following is the parent element of the **ChartMember.DataElementName** element.

Parent elements
<a href="#">ChartMember</a>

The following is the XML Schema definition of the **ChartMember.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

#### 2.150.4 ChartMember.DataElementOutput

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartMember.DataElementOutput** element specifies whether an item **MUST** appear in a **data rendering**. This element is optional.

If the **ChartMember.DataElementOutput** element is present, its value **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**Output:** Specifies that the item appears in the data rendering output.

**NoOutput:** Specifies that the item does not appear in the output.

**Auto:** If the [ChartMember.Group](#) element is present, the value of this element is interpreted as "Output".

If the **ChartMember.DataElementOutput** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartMember.DataElementOutput** element.

Parent elements
<a href="#">ChartMember</a>

The following is the XML Schema definition of the **ChartMember.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0" maxOccurs="1">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

#### 2.150.5 ChartMember.Group

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartMember.Group** element specifies the expressions by which to group data. This element is optional. If this element is not present, it is a **static member**. Otherwise, this element is a **dynamic member**. If there are any [ChartMember](#) elements with **detail group** among the ancestors of the parent element of this element, this element **MUST NOT** be specified.

This element is of type **Group**.

Following is the parent element of the **ChartMember.Group** element.



Parent elements
<a href="#">ChartMember</a>

The following is the XML Schema definition of the **ChartMember.Group** element.

```
<xsd:element name="Group" type="GroupType" minOccurs="0" maxOccurs="1" />
```

## 2.150.6 ChartMember.Label

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

If the value of the [ChartSeries.Type](#) element not is "Shape", the **ChartMember.Label** element specifies the label to display in the legend for [ChartMembers](#) in the [Chart.ChartSeriesHierarchy](#), and it specifies the labels to display on the **CategoryAxis** for **ChartMembers** in the [Chart.ChartCategoryHierarchy](#).

If the value of the **ChartSeries.Type** element is "Shape", the **ChartMember.Label** element specifies the label to display in the legend for **ChartMembers** in the **Chart.ChartCategoryHierarchy**.

The **ChartMember.Label** element MUST be specified, and its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **ChartMember.Label** element.

Parent elements
<a href="#">ChartMember</a>

The following is the XML Schema definition of the **ChartMember.Label** element.

```
<xsd:element name="Label" type="StringLocIDType" minOccurs="1" maxOccurs="1" />
```

## 2.150.7 ChartMember.SortExpressions

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartMember.SortExpressions** element specifies the expressions by which to sort member instances. This element is optional. This element MUST NOT be specified if the value of the [ChartMember.Group](#) element is not specified. The **ChartMember.SortExpressions** element is of type [SortExpressions](#).

Following is the parent element of the **ChartMember.SortExpressions** element.

Parent elements
<a href="#">ChartMember</a>

The following is the XML Schema definition of the **ChartMember.SortExpressions** element.

```
<xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" maxOccurs="1" />
```

## 2.151 ChartLegends

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegends** element specifies an ordered list of [ChartLegend](#) elements for a [Chart](#).

The following are the parent and child elements of the **ChartLegends** element.

Parent elements
Chart

Child elements
<a href="#">ChartLegends.ChartLegend</a>

The following is the XML Schema definition of the **ChartLegends** element in RDL 2008/01.

```
<xsd:complexType name="ChartLegendsType">
  <xsd:sequence>
    <xsd:element name="ChartLegend" type="ChartLegendType" minOccurs="0"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartLegends** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartLegendsType">
  <xsd:sequence>
    <xsd:element name="ChartLegend" type="ChartLegendType" minOccurs="0"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.151.1 ChartLegends.ChartLegend

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegends.ChartLegend** element specifies a [ChartLegend](#) in the [ChartLegends](#) of a [Chart](#). The **ChartLegends.ChartLegend** element is optional. This element is of type **ChartLegend**.

Following is the parent element of the **ChartLegends.ChartLegend** element.

Parent elements
ChartLegends

The following is the XML Schema definition of the **ChartLegends.ChartLegend** element.

```
<xsd:element name="ChartLegend" type="ChartLegendType" minOccurs="0"
  maxOccurs="unbounded" />
```

## 2.152 ChartLegend

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegend** element specifies the properties that can be used to display instances of a series group in a legend in a [Chart](#).

The following are the parent element, attribute, and child elements of the **ChartLegend** element.

Parent elements
<a href="#">ChartLegends</a>

Attributes
<a href="#">ChartLegend.Name</a>

Child elements
<a href="#">ChartLegend.AutoFitTextDisabled</a>
<a href="#">ChartLegend.ChartElementPosition</a>
<a href="#">ChartLegend.ChartLegendColumns</a>
<a href="#">ChartLegend.ChartLegendTitle</a>
<a href="#">ChartLegend.ColumnSeparator</a>
<a href="#">ChartLegend.ColumnSeparatorColor</a>
<a href="#">ChartLegend.ColumnSpacing</a>
<a href="#">ChartLegend.DockOutsideChartArea</a>
<a href="#">ChartLegend.DockToChartArea</a>
<a href="#">ChartLegend.EquallySpacedItems</a>
<a href="#">ChartLegend.HeaderSeparator</a>
<a href="#">ChartLegend.HeaderSeparatorColor</a>
<a href="#">ChartLegend.Hidden</a>
<a href="#">ChartLegend.InterlacedRows</a>
<a href="#">ChartLegend.InterlacedRowsColor</a>
<a href="#">ChartLegend.Layout</a>
<a href="#">ChartLegend.MaxAutoSize</a>
<a href="#">ChartLegend.MinFontSize</a>
<a href="#">ChartLegend.Position</a>
<a href="#">ChartLegend.Reversed</a>

Child elements
<a href="#">ChartLegend.Style</a>
<a href="#">ChartLegend.TextWrapThreshold</a>

The following is the XML Schema definition of the **ChartLegend** element in RDL 2008/01.

```
<xsd:complexType name="ChartLegendType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="Layout" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockToChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartElementPosition" type="ChartElementPositionType"
      minOccurs="0" />
    <xsd:element name="ChartLegendTitle" type="ChartLegendTitleType"
      minOccurs="0" />
    <xsd:element name="AutoFitTextDisabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartLegendColumns" type="ChartLegendColumnsType"
      minOccurs="0" />
    <xsd:element name="HeaderSeparator" type="xsd:string" minOccurs="0" />
    <xsd:element name="HeaderSeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ColumnSeparator" type="xsd:string" minOccurs="0" />
    <xsd:element name="ColumnSeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ColumnSpacing" type="xsd:string" minOccurs="0" />
    <xsd:element name="InterlacedRows" type="xsd:string" minOccurs="0" />
    <xsd:element name="InterlacedRowsColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="EquallySpacedItems" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaxAutoSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextWrapThreshold" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartLegend** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartLegendType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="Layout" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockToChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartElementPosition" type="ChartElementPositionType"
      minOccurs="0" />
    <xsd:element name="ChartLegendTitle" type="ChartLegendTitleType"
      minOccurs="0" />
    <xsd:element name="AutoFitTextDisabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartLegendColumns" type="ChartLegendColumnsType"
      minOccurs="0" />
    <xsd:element name="HeaderSeparator" type="xsd:string" minOccurs="0" />
    <xsd:element name="HeaderSeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ColumnSeparator" type="xsd:string" minOccurs="0" />
    <xsd:element name="ColumnSeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ColumnSpacing" type="xsd:string" minOccurs="0" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

```

<xsd:element name="InterlacedRows" type="xsd:string" minOccurs="0" />
<xsd:element name="InterlacedRowsColor" type="xsd:string" minOccurs="0" />
<xsd:element name="EquallySpacedItems" type="xsd:string" minOccurs="0" />
<xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
<xsd:element name="MaxAutoSize" type="xsd:string" minOccurs="0" />
<xsd:element name="TextWrapThreshold" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.152.1 ChartLegend.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegend.Name** attribute specifies the name of a [ChartLegend](#). This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique among the **ChartLegend.Name** values in the parent collection.

Following is the parent element of the **ChartLegend.Name** attribute.

Parent elements
<a href="#">ChartLegend</a>

The following is the XML Schema definition of the **ChartLegend.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.152.2 ChartLegend.AutoFitTextDisabled

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegend.AutoFitTextDisabled** element specifies whether text will be autosized to fit in the legend area. The **ChartLegend.AutoFitTextDisabled** element is optional. If this element is present, its value **MUST** be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartLegend.AutoFitTextDisabled** element.

Parent elements
<a href="#">ChartLegend</a>

The following is the XML Schema definition of the **ChartLegend.AutoFitTextDisabled** element.

```
<xsd:element name="AutoFitTextDisabled" type="xsd:string" minOccurs="0" />
```

### 2.152.3 ChartLegend.ChartElementPosition

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegend.ChartElementPosition** element specifies a custom position for a [ChartLegend](#). The **ChartLegend.ChartElementPosition** element is optional. If this element is not present, automatic positioning is used. This element is of type [ChartElementPosition](#).

Following is the parent element of the **ChartLegend.ChartElementPosition** element.

Parent elements
<a href="#">ChartLegend</a>

The following is the XML Schema definition of the **ChartLegend.ChartElementPosition** element.

```
<xsd:element name="ChartElementPosition" type="ChartElementPositionType"
  minOccurs="0" />
```

#### 2.152.4 ChartLegend.ChartLegendColumns

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.ChartLegendColumns** element is ignored. This element is of type [ChartLegendColumns](#).

Following is the parent element of the **ChartLegend.ChartLegendColumns** element.

Parent elements
<a href="#">ChartLegend</a>

The following is the XML Schema definition of the **ChartLegend.ChartLegendColumns** element.

```
<xsd:element name="ChartLegendColumns" type="ChartLegendColumnsType"
  minOccurs="0" />
```

#### 2.152.5 ChartLegend.ChartLegendTitle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.ChartLegendTitle** element specifies the title of a [ChartLegend](#). This element is optional. This element is of type [ChartLegendTitle](#).

Following is the parent element of the **ChartLegend.ChartLegendTitle** element.

Parent elements
<a href="#">ChartLegend</a>

The following is the XML Schema definition of the **ChartLegend.ChartLegendTitle** element.

```
<xsd:element name="ChartLegendTitle" type="ChartLegendTitleType" minOccurs="0" />
```

#### 2.152.6 ChartLegend.ColumnSeparator

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.ColumnSeparator** element specifies what type of separator MUST be used for columns in a [ChartLegend](#). This element is optional.

If the **ChartLegend.ColumnSeparator** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**None:** Specifies no separator.

**Line:** Specifies to separate with a line.

**ThickLine:** Specifies to separate with a thick line.

**DoubleLine:** Specifies to separate with a double line.

**DashLine:** Specifies to separate with a dashed line.

**DotLine:** Specifies to separate with a dotted line.

**GradientLine:** Specifies to separate with a gradient line.

**ThickGradientLine:** Specifies to separate with a thick gradient line.

If the **ChartLegend.ColumnSeparator** element is not present, its value is interpreted as "None".

Following is the parent element of the **ChartLegend.ColumnSeparator** element.

Parent elements
ChartLegend

The following is the XML Schema definition of the **ChartLegend.ColumnSeparator** element.

```
<xsd:element name="ColumnSeparator" type="xsd:string" minOccurs="0" />
```

## 2.152.7 ChartLegend.ColumnSeparatorColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.ColumnSeparatorColor** element specifies the color of a column separator in a [ChartLegend](#). The **ChartLegend.ColumnSeparatorColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. If this element is not present, its value is interpreted as an empty color.

Following is the parent element of the **ChartLegend.ColumnSeparatorColor** element.

Parent elements
ChartLegend

The following is the XML Schema definition of the **ChartLegend.ColumnSeparatorColor** element.

```
<xsd:element name="ColumnSeparatorColor" type="xsd:string" minOccurs="0" />
```

## 2.152.8 ChartLegend.ColumnSpacing

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.ColumnSpacing** element specifies the spacing between legend columns as a percentage of the font size of the legend (as specified by [ChartLegend.Style](#)). The **ChartLegend.ColumnSpacing** element is optional.

If the **ChartLegend.ColumnSpacing** element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 50.

The value of this element MUST be greater than or equal to 0. <32>

Following is the parent element of the **ChartLegend.ColumnSpacing** element.

Parent elements
<a href="#">ChartLegend</a>

The following is the XML Schema definition of the **ChartLegend.ColumnSpacing** element.

```
<xsd:element name="ColumnSpacing" type="xsd:string" minOccurs="0" />
```

### 2.152.9 ChartLegend.DockOutsideChartArea

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegend.DockOutsideChartArea** element specifies whether a [ChartLegend](#) MUST be docked outside a [ChartArea](#). This element is ignored if [ChartLegend.DockToChartArea](#) is not specified.

The **ChartLegend.DockOutsideChartArea** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartLegend.DockOutsideChartArea** element.

Parent elements
<a href="#">ChartLegend</a>

The following is the XML Schema definition of the **ChartLegend.DockOutsideChartArea** element.

```
<xsd:element name="DockOutsideChartArea" type="xsd:string" minOccurs="0" />
```

### 2.152.10 ChartLegend.DockToChartArea

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegend.DockToChartArea** element specifies the name of the [ChartArea](#) on which to draw a [ChartLegend](#). The **ChartLegend.DockToChartArea** element is optional. If the **ChartLegend.DockToChartArea** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1). If this element is not specified or if it does not match any **ChartArea** name, the legend MUST be drawn relative to the chart area.

Following is the parent element of the **ChartLegend.DockToChartArea** element.

Parent elements
<a href="#">ChartLegend</a>



The following is the XML Schema definition of the **ChartLegend.DockToChartArea** element.

```
<xsd:element name="DockToChartArea" type="xsd:string" minOccurs="0" />
```

### 2.152.11 ChartLegend.EquallySpacedItems

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.EquallySpacedItems** element specifies whether chart legend items MUST be equally spaced. The **ChartLegend.EquallySpacedItems** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartLegend.EquallySpacedItems** element.

Parent elements
<a href="#">ChartLegend</a>

The following is the XML Schema definition of the **ChartLegend.EquallySpacedItems** element.

```
<xsd:element name="EquallySpacedItems" type="xsd:string" minOccurs="0" />
```

### 2.152.12 ChartLegend.HeaderSeparator

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.HeaderSeparator** element specifies the type of separator to use for the chart legend header in a [Chart](#). This element is optional.

If the **ChartLegend.HeaderSeparator** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**None:** Specifies no separator.

**Line:** Specifies to separate with a line.

**ThickLine:** Specifies to separate with thick line.

**DoubleLine:** Specifies to separate with a double line.

**DashLine:** Specifies to separate with a dashed line.

**DotLine:** Specifies to separate with a dotted line.

**GradientLine:** Specifies to separate with a gradient line.

**ThickGradientLine:** Specifies to separate with a thick gradient line.

If the **ChartLegend.HeaderSeparator** element is not present, its value is interpreted as "None".

Following is the parent element of the **ChartLegend.HeaderSeparator** element.

Parent elements
<a href="#">ChartLegend</a>

The following is the XML Schema definition of the **ChartLegend.HeaderSeparator** element.

```
<xsd:element name="HeaderSeparator" type="xsd:string" minOccurs="0" />
```

### 2.152.13 ChartLegend.HeaderSeparatorColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.HeaderSeparatorColor** element specifies the color of a chart legend header separator. This element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. If this element is not present, its value is interpreted as an empty color.

Following is the parent element of the **ChartLegend.HeaderSeparatorColor** element.

Parent elements
<a href="#">ChartLegend</a>

The following is the XML Schema definition of the **ChartLegend.HeaderSeparatorColor** element.

```
<xsd:element name="HeaderSeparatorColor" type="xsd:string" minOccurs="0" />
```

### 2.152.14 ChartLegend.Hidden

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.Hidden** element specifies whether a [ChartLegend](#) is hidden. The **ChartLegend.Hidden** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartLegend.Hidden** element.

Parent elements
<a href="#">ChartLegend</a>

The following is the XML Schema definition of the **ChartLegend.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

### 2.152.15 ChartLegend.InterlacedRows

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.InterlacedRows** element specifies whether legend rows use interlaced colors. The **ChartLegend.InterlacedRows** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartLegend.InterlacedRows** element.

Parent elements
-----------------

<a href="#">ChartLegend</a>
-----------------------------

The following is the XML Schema definition of the **ChartLegend.InterlacedRows** element.

```
<xsd:element name="InterlacedRows" type="xsd:string" minOccurs="0" />
```

## 2.152.16 ChartLegend.InterlacedRowsColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.InterlacedRowsColor** element specifies the background color to use for interlaced rows in a [ChartLegend](#). The **ChartLegend.InterlacedRowsColor** element is optional. The value of this element MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. If the **ChartLegend.InterlacedRowsColor** element is not present, the background color of the [ChartArea](#) is used, as specified in the [ChartArea.Style](#) element.

Following is the parent element of the **ChartLegend.InterlacedRowsColor** element.

Parent elements
-----------------

<a href="#">ChartLegend</a>
-----------------------------

The following is the XML Schema definition of the **ChartLegend.InterlacedRowsColor** element.

```
<xsd:element name="InterlacedRowsColor" type="xsd:string" minOccurs="0" />
```

## 2.152.17 ChartLegend.Layout

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.Layout** element specifies the arrangement of labels in a [ChartLegend](#). This element is optional.

If the **ChartLegend.Layout** element is present, its value MUST be a [String](#) ([IXMLSCHEMA2/21](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**AutoTable**: Specifies that labels are arranged automatically to fit.

**Column**: Specifies that labels are arranged in a column.

**Row**: Specifies that labels are arranged in a row.

**WideTable**: Specifies that labels are arranged in a wide table.

**TallTable**: Specifies that labels are arranged in a tall table.

If the **ChartLegend.Layout** element is not present, its value is interpreted as "AutoTable".

Following is the parent element of the **ChartLegend.Layout** element.

Parent elements
-----------------

<a href="#">ChartLegend</a>
-----------------------------

The following is the XML Schema definition of the **ChartLegend.Layout** element.

```
<xsd:element name="Layout" type="xsd:string" minOccurs="0" />
```

### 2.152.18 ChartLegend.MaxAutoSize

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.MaxAutoSize** element specifies the maximum size for a [ChartLegend](#) as a percentage of the [Chart](#) size. This element is optional.

If the **ChartLegend.MaxAutoSize** element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 50. The value of this element MUST be greater than or equal to 0 and less than or equal to 100.

Following is the parent element of the **ChartLegend.MaxAutoSize** element.

Parent elements
ChartLegend

The following is the XML Schema definition of the **ChartLegend.MaxAutoSize** element.

```
<xsd:element name="MaxAutoSize" type="xsd:string" minOccurs="0" />
```

### 2.152.19 ChartLegend.MinFontSize

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.MinFontSize** element specifies the minimum font size for the algorithm that auto-sizes the text in a [ChartLegend](#). To use this element, the [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) value of [ChartLegend.AutoFitTextDisabled](#) MUST be false.

The **ChartLegend.MinFontSize** element is optional. If this element is present, its value MUST be an [RdSize](#) or an expression that evaluates to an **RdSize**. If this element is not present, its value is interpreted as 7pt. The value of this element MUST be at least 5pt.

Following is the parent element of the **ChartLegend.MinFontSize** element.

Parent elements
ChartLegend

The following is the XML Schema definition of the **ChartLegend.MinFontSize** element.

```
<xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
```

### 2.152.20 ChartLegend.Position

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegend.Position** element specifies the position of a [ChartLegend](#) relative to the sides and corners of the container ([Chart](#) or [ChartArea](#)). This element is optional.

If the **ChartLegend.Position** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**RightTop**: Positions the legend at the upper corner of the right side of the chart.

**TopLeft**: Positions the legend at the left corner of the upper side of the chart.

**TopCenter**: Positions the legend at the middle of the upper side of the chart.

**TopRight**: Positions the legend at the right corner of the upper side of the chart.

**LeftTop**: Positions the legend at the upper corner of the left side of the chart.

**LeftCenter**: Positions the legend at the middle of the left side of the chart.

**LeftBottom**: Positions the legend at the bottom corner of the left side of the chart.

**RightCenter**: Positions the legend at the middle of the right side of the chart.

**RightBottom**: Positions the legend at the bottom corner of the right side of the chart.

**BottomRight**: Positions the legend at the right corner of the bottom side of the chart.

**BottomCenter**: Positions the legend at the middle of the bottom side of the chart.

**BottomLeft**: Positions the legend at the left corner of the bottom side of the chart.

If the **ChartLegend.Position** element is not present, its value is interpreted as "RightTop".

Following is the parent element of the **ChartLegend.Position** element.

Parent elements
ChartLegend

The following is the XML Schema definition of the **ChartLegend.Position** element.

```
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
```

## 2.152.21 ChartLegend.Reversed

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegend.Reversed** element specifies whether the direction (the order of the legend contents in each row) of a [ChartLegend](#) is to be reversed. This element is optional.

If the **ChartLegend.Reversed** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Auto**: Specifies that the direction is automatically detected based on the series types (specified by [ChartSeries.Type](#) and [ChartSeries.Subtype](#)).

**True**: Specifies that the order of items in the legend is to be reversed.

**False**: Specifies that the order of items in the legend is not to be reversed.

If the **ChartLegend.Reversed** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartLegend.Reversed** element.

Parent elements
ChartLegend

The following is the XML Schema definition of the **ChartLegend.Reversed** element.

```
<xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
```

### 2.152.22 ChartLegend.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegend.Style** element specifies the style properties for a [ChartLegend](#). This element is optional. This element is of type [Style](#).

Following is the parent element of the **ChartLegend.Style** element.

Parent elements
ChartLegend

The following is the XML Schema definition of the **ChartLegend.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.152.23 ChartLegend.TextWrapThreshold

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegend.TextWrapThreshold** element specifies the number of characters after which to wrap text in a [ChartLegend](#).

The **ChartLegend.TextWrapThreshold** element is optional. If this element is present, its value MUST be a non-negative [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to a non-negative **Integer**. If this element is not present, its value is interpreted as 25.

Following is the parent element of the **ChartLegend.TextWrapThreshold** element.

Parent elements
ChartLegend

The following is the XML Schema definition of the **ChartLegend.TextWrapThreshold** element.

```
<xsd:element name="TextWrapThreshold" type="xsd:string" minOccurs="0" />
```

### 2.153 ChartLegend.Columns

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegend.Columns** element is ignored.

The following are the parent and child elements of the **ChartLegendColumns** element.

Parent elements
<a href="#">ChartLegend</a>

Child elements
<a href="#">ChartLegendColumns.ChartLegendColumn</a>

The following is the XML Schema definition of the **ChartLegendColumns** element in RDL 2008/01.

```
<xsd:complexType name="ChartLegendColumnsType">
  <xsd:sequence>
    <xsd:element name="ChartLegendColumn" type="ChartLegendColumnType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartLegendColumns** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartLegendColumnsType">
  <xsd:sequence>
    <xsd:element name="ChartLegendColumn" type="ChartLegendColumnType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.153.1 ChartLegendColumns.ChartLegendColumn

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendColumns.ChartLegendColumn** element is ignored. This element is of type [ChartLegendColumn](#).

Following is the parent element of the **ChartLegendColumns.ChartLegendColumn** element.

Parent elements
<a href="#">ChartLegendColumns</a>

The following is the XML Schema definition of the **ChartLegendColumns.ChartLegendColumn** element.

```
<xsd:element name="ChartLegendColumn" type="ChartLegendColumnType"
  maxOccurs="unbounded" />
```

### 2.154 ChartLegendColumn

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendColumn** element is ignored.

The following are the parent elements, attributes, and child elements of the **ChartLegendColumn** element.

Parent elements
<a href="#">ChartLegendColumns</a>

Attributes
<a href="#">ChartLegendColumn.Name</a>

Child elements
<a href="#">ChartLegendColumn.ActionInfo</a>
<a href="#">ChartLegendColumn.ColumnType</a>
<a href="#">ChartLegendColumn.MaximumWidth</a>
<a href="#">ChartLegendColumn.MinimumWidth</a>
<a href="#">ChartLegendColumn.SeriesSymbolHeight</a>
<a href="#">ChartLegendColumn.SeriesSymbolWidth</a>
<a href="#">ChartLegendColumn.Style</a>
<a href="#">ChartLegendColumn.ToolTip</a>
<a href="#">ChartLegendColumn.Value</a>

The following is the XML Schema definition of the **ChartLegendColumn** element in RDL 2008/01.

```
<xsd:complexType name="ChartLegendColumnType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ColumnType">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Text" />
          <xsd:enumeration value="SeriesSymbol" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="MinimumWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeriesSymbolWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeriesSymbolHeight" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```



The following is the XML Schema definition of the **ChartLegendColumn** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartLegendColumnType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ColumnType">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Text" />
          <xsd:enumeration value="SeriesSymbol" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="MinimumWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeriesSymbolWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeriesSymbolHeight" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.154.1 ChartLegendColumn.Name

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendColumn.Name** attribute specifies the name of the [ChartLegendColumn](#). This attribute MUST be present. Its value MUST be a **CLS-compliant identifier** [UTR15] that is unique among the **ChartLegendColumn.Name** values in the parent collection.

Following is the parent element of the **ChartLegendColumn.Name** attribute.

Parent elements
<a href="#">ChartLegendColumn</a>

The following is the XML Schema definition of the **ChartLegendColumn.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.154.2 ChartLegendColumn.ActionInfo

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendColumn.ActionInfo** element is ignored. This element is of type [ActionInfo](#).

Following is the parent element of the **ChartLegendColumn.ActionInfo** element.

Parent elements
<a href="#">ChartLegendColumn</a>

The following is the XML Schema definition of the **ChartLegendColumn.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.154.3 ChartLegendColumn.ColumnType

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendColumn.ColumnType** element is ignored.

Following is the parent element of the **ChartLegendColumn.ColumnType** element.

Parent elements
<a href="#">ChartLegendColumn</a>

The following is the XML Schema definition of the **ChartLegendColumn.ColumnType** element.

```
<xsd:element name="ColumnType">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Text" />
      <xsd:enumeration value="SeriesSymbol" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.154.4 ChartLegendColumn.MaximumWidth

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendColumn.MaximumWidth** element is ignored.

Following is the parent element of the **ChartLegendColumn.MaximumWidth** element.

Parent elements
<a href="#">ChartLegendColumn</a>

The following is the XML Schema definition of the **ChartLegendColumn.MaximumWidth** element.

```
<xsd:element name="MaximumWidth" type="xsd:string" minOccurs="0" />
```

### 2.154.5 ChartLegendColumn.MinimumWidth

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendColumn.MinimumWidth** element is ignored.

Parent elements
<a href="#">ChartLegendColumn</a>

The following is the XML Schema definition of the **ChartLegendColumn.MinimumWidth** element.

```
<xsd:element name="MinimumWidth" type="xsd:string" minOccurs="0" />
```

## 2.154.6 ChartLegendColumn.SeriesSymbolHeight

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendColumn.SeriesSymbolHeight** element is ignored.

Following is the parent element of the **ChartLegendColumn.SeriesSymbolHeight** element.

Parent elements
<a href="#">ChartLegendColumn</a>

The following is the XML Schema definition of the **ChartLegendColumn.SeriesSymbolHeight** element.

```
<xsd:element name="SeriesSymbolHeight" type="xsd:string" minOccurs="0" />
```

## 2.154.7 ChartLegendColumn.SeriesSymbolWidth

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendColumn.SeriesSymbolWidth** element is ignored.

Following is the parent element of the **ChartLegendColumn.SeriesSymbolWidth** element.

Parent elements
<a href="#">ChartLegendColumn</a>

The following is the XML Schema definition of the **ChartLegendColumn.SeriesSymbolWidth** element.

```
<xsd:element name="SeriesSymbolWidth" type="xsd:string" minOccurs="0" />
```

## 2.154.8 ChartLegendColumn.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendColumn.Style** element is ignored. This element is of type [Style](#).

Following is the parent element of the **ChartLegendColumn.Style** element.

Parent elements
<a href="#">ChartLegendColumn</a>

The following is the XML Schema definition of the **ChartLegendColumn.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.154.9 ChartLegendColumn.ToolTip

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendColumn.ToolTip** element is ignored.

Following is the parent element of the **ChartLegendColumn.ToolTip** element.

Parent elements
<a href="#">ChartLegendColumn</a>

The following is the XML Schema definition of the **ChartLegendColumn.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

## 2.154.10 ChartLegendColumn.Value

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendColumn.Value** element is ignored.

Following is the parent element of the **ChartLegendColumn.Value** element.

Parent elements
<a href="#">ChartLegendColumn</a>

The following is the XML Schema definition of the **ChartLegendColumn.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="0" />
```

## 2.155 ChartLegendTitle

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendTitle** element specifies a title for a legend.

The following are the parent and child elements of the **ChartLegendTitle** element.

Parent elements
<a href="#">ChartLegend</a>

Child elements
<a href="#">ChartLegendTitle.Caption</a>

<b>Child elements</b>
<a href="#">ChartLegendTitle.Style</a>
<a href="#">ChartLegendTitle.TitleSeparator</a>

The following is the XML Schema definition of the **ChartLegendTitle** element in RDL 2008/01.

```
<xsd:complexType name="ChartLegendTitleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Caption" type="StringLocIDType" />
    <xsd:element name="TitleSeparator" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartLegendTitle** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartLegendTitleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Caption" type="StringLocIDType" />
    <xsd:element name="TitleSeparator" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.155.1 ChartLegendTitle.Caption

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendTitle.Caption** element specifies the caption of a chart legend title. This element **MUST** be specified. The value of this element **MUST** be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The **String** value can be an empty string. Following is the parent element of the **ChartLegendTitle.Caption** element.

<b>Parent elements</b>
<a href="#">ChartLegendTitle</a>

The following is the XML Schema definition of the **ChartLegendTitle.Caption** element.

```
<xsd:element name="Caption" type="StringLocIDType" />
```

### 2.155.2 ChartLegendTitle.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendTitle.Style** element specifies the style properties for a chart legend title. This element is optional. This element is of type [Style](#).

Following is the parent element of the **ChartLegendTitle.Style** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartLegendTitle</a>
----------------------------------

The following is the XML Schema definition of the **ChartLegendTitle.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.155.3 ChartLegendTitle.TitleSeparator

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendTitle.TitleSeparator** element specifies the type of separator to use for a chart legend title. This element is optional.

If the **ChartLegendTitle.TitleSeparator** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**None**: Specifies no separator.

**Line**: Specifies to separate with a line.

**ThickLine**: Specifies to separate with a thick line.

**DoubleLine**: Specifies to separate with a double line.

**DashLine**: Specifies to separate with a dashed line.

**Dotline**: Specifies to separate with a dotted line.

**GradientLine**: Specifies to separate with a gradient line.

**ThickGradientLine**: Specifies to separate with a thick gradient line.

If the **ChartLegendTitle.TitleSeparator** element is not present, its value is interpreted as "None".

Following is the parent element of the **ChartLegendTitle.TitleSeparator** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartLegendTitle</a>
----------------------------------

The following is the XML Schema definition of the **ChartLegendTitle.TitleSeparator** element.

```
<xsd:element name="TitleSeparator" type="xsd:string" minOccurs="0" />
```

### 2.156 ChartTitles

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTitles** element specifies an ordered list of [ChartTitle](#) elements for a chart.

The following are the parent elements and child element of the **ChartTitles** element.

Parent elements
<a href="#">Chart</a>

Child elements
<a href="#">ChartTitles.ChartTitle</a>

The following is the XML Schema definition of the **ChartTitles** element in RDL 2008/01.

```
<xsd:complexType name="ChartTitlesType">
  <xsd:sequence>
    <xsd:element name="ChartTitle" type="ChartTitleType" minOccurs="0"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartTitles** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartTitlesType">
  <xsd:sequence>
    <xsd:element name="ChartTitle" type="ChartTitleType" minOccurs="0"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.156.1 ChartTitles.ChartTitle

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTitles.ChartTitle** element specifies a [ChartTitle](#) element within the [ChartTitles](#) element for a [Chart](#). The **ChartTitles.ChartTitle** element is optional. This element is of type **ChartTitle**.

Following is the parent element of the **ChartTitles.ChartTitle** element.

Parent elements
<a href="#">ChartTitles</a>

The following is the XML Schema definition of the **ChartTitles.ChartTitle** element.

```
<xsd:element name="ChartTitle" type="ChartTitleType" minOccurs="0" maxOccurs="unbounded" />
```

### 2.157 ChartTitle

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTitle** element specifies the properties for a ChartTitle in a [Chart](#).

The following are the parent elements, attributes, and child elements of the **ChartTitle** element.

Parent elements
Chart
<a href="#">ChartTitles</a>

Attributes
<a href="#">ChartTitle.Name</a>

Child elements
<a href="#">ChartTitle.ActionInfo</a>
<a href="#">ChartTitle.Caption</a>
<a href="#">ChartTitle.ChartElementPosition</a>
<a href="#">ChartTitle.DockOffset</a>
<a href="#">ChartTitle.DockOutsideChartArea</a>
<a href="#">ChartTitle.DockToChartArea</a>
<a href="#">ChartTitle.Hidden</a>
<a href="#">ChartTitle.Position</a>
<a href="#">ChartTitle.Style</a>
<a href="#">ChartTitle.TextOrientation</a>
<a href="#">ChartTitle.ToolTip</a>

The following is the XML Schema definition of the **ChartTitle** element in RDL 2008/01.

```

<xsd:complexType name="ChartTitleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Caption" type="StringLocIDType" minOccurs="1" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockToChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartElementPosition" type="ChartElementPositionType"
      minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **ChartTitle** element in RDL 2010/01 and RDL 2016/01.



```

<xsd:complexType name="ChartTitleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Caption" type="StringLocIDType" minOccurs="1" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockToChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartElementPosition" type="ChartElementPositionType"
      minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

## 2.157.1 ChartTitle.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTitle.Name** attribute specifies the name of a [ChartTitle](#). This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique among the **ChartTitle.Name** values in the parent collection.

Following is the parent element of the **ChartTitle.Name** attribute.

Parent elements
ChartTitle

The following is the XML Schema definition of the **ChartTitle.Name** attribute.

```

<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />

```

## 2.157.2 ChartTitle.ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTitle.ActionInfo** element specifies actions that are associated with a [ChartTitle](#). This element is optional. The **ChartTitle.ActionInfo** element is of type [ActionInfo](#).

Following is the parent element of the **ChartTitle.ActionInfo** element.

Parent elements
ChartTitle

The following is the XML Schema definition of the **ChartTitle.ActionInfo** element.

```

<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />

```

### 2.157.3 ChartTitle.Caption

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTitle.Caption** element specifies the caption of a [ChartTitle](#). This element MUST be specified, and its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **ChartTitle.Caption** element.

Parent elements
<a href="#">ChartTitle</a>

The following is the XML Schema definition of the **ChartTitle.Caption** element.

```
<xsd:element name="Caption" type="StringLocIDType" minOccurs="1" />
```

### 2.157.4 ChartTitle.ChartElementPosition

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTitle.ChartElementPosition** element specifies a custom position for a [ChartTitle](#). This element is optional. To enable automatic positioning, this element MUST be omitted. The **ChartTitle.ChartElementPosition** element is of type [ChartElementPosition](#).

Following is the parent element of the **ChartTitle.ChartElementPosition** element.

Parent elements
<a href="#">ChartTitle</a>

The following is the XML Schema definition of the **ChartTitle.ChartElementPosition** element.

```
<xsd:element name="ChartElementPosition" type="ChartElementPositionType" minOccurs="0" />
```

### 2.157.5 ChartTitle.DockOffset

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTitle.DockOffset** element specifies the offset from the dock location as a percentage of the size of the [Chart](#). This element is optional. If the **ChartTitle.DockOffset** element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **ChartTitle.DockOffset** element.

Parent elements
<a href="#">ChartTitle</a>

The following is the XML Schema definition of the **ChartTitle.DockOffset** element.

```
<xsd:element name="DockOffset" type="xsd:string" minOccurs="0" />
```

## 2.157.6 ChartTitle.DockOutsideChartArea

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTitle.DockOutsideChartArea** element specifies whether a **ChartTitle** is docked outside a [ChartArea](#) element. The **ChartTitle.DockOutsideChartArea** element is optional. To use this element, the [ChartTitle.DockToChartArea](#) element MUST be set.

If the **ChartTitle.DockOutsideChartArea** element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartTitle.DockOutsideChartArea** element.

Parent elements
<a href="#">ChartTitle</a>

The following is the XML Schema definition of the **ChartTitle.DockOutsideChartArea** element.

```
<xsd:element name="DockOutsideChartArea" type="xsd:string" minOccurs="0" />
```

## 2.157.7 ChartTitle.DockToChartArea

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTitle.DockToChartArea** element specifies the name of a [ChartArea](#) element on which to draw a [ChartTitle](#). The **ChartTitle.DockToChartArea** element is optional. If the **ChartTitle.DockToChartArea** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1). If this element is omitted or does not match any chart area name, the title MUST be drawn relative to the [Chart](#).

Following is the parent element of the **ChartTitle.DockToChartArea** element.

Parent elements
<a href="#">ChartTitle</a>

The following is the XML Schema definition of the **ChartTitle.DockToChartArea** element.

```
<xsd:element name="DockToChartArea" type="xsd:string" minOccurs="0" />
```

## 2.157.8 ChartTitle.Hidden

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartTitle.Hidden** element specifies whether a [ChartTitle](#) is hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ChartTitle.Hidden** element.

Parent elements
ChartTitle

The following is the XML Schema definition of the **ChartTitle.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

## 2.157.9 ChartTitle.Position

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartTitle.Position** element specifies the position of a [ChartTitle](#). This element is optional. If the **ChartTitle.Position** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/21](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**TopCenter**: Specifies that the title is positioned at the top-center of the chart.

**TopLeft**: Specifies that the title is positioned at the top-left of the chart.

**TopRight**: Specifies that the title is positioned at the top-right of the chart.

**LeftTop**: Specifies that the title is positioned to the left-top of the chart.

**LeftCenter**: Specifies that the title is positioned to the left-center of the chart.

**LeftBottom**: Specifies that the title is positioned at to the left-bottom of the chart.

**RightTop**: Specifies that the title is positioned at to the right-top of the chart.

**RightCenter**: Specifies that the title is positioned to the right-center of the chart.

**RightBottom**: Specifies that the title is positioned to the right-bottom of the chart.

**BottomRight**: Specifies that the title is positioned at the bottom-right of the chart.

**BottomCenter**: Specifies that the title is positioned at the bottom-center of the chart.

**BottomLeft**: Specifies that the title is positioned at the bottom-left of the chart.

If the **ChartTitle.Position** element is not present, its value is interpreted as "TopCenter".

Following is the parent element of the **ChartTitle.Position** element.

Parent elements
ChartTitle

The following is the XML Schema definition of the **ChartTitle.Position** element.

```
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
```

## 2.157.10 ChartTitle.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartTitle.Style** element specifies the style properties for a [ChartTitle](#). This element is optional. The **ChartTitle.Style** element is of type [Style](#).

Following is the parent element of the **ChartTitle.Style** element.

Parent elements
ChartTitle

The following is the XML Schema definition of the **ChartTitle.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.157.11 ChartTitle.TextOrientation

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartTitle.TextOrientation** element specifies the orientation of the text of a [ChartTitle](#). This element is optional. If the **ChartTitle.TextOrientation** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Auto**: Specifies whether the orientation is selected automatically based on context (for example, Rotated270 for titles that are docked on the left).

**Horizontal**: Specifies horizontal text.

**Rotated90**: Specifies vertical text, rotated 90 degrees.

**Rotated270**: Specifies vertical text, rotated 270 degrees.

**Stacked**: Specifies vertical text with no character rotation.

If the **ChartTitle.TextOrientation** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **ChartTitle.TextOrientation** element.

Parent elements
ChartTitle

The following is the XML Schema definition of the **ChartTitle.TextOrientation** element.

```
<xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
```

### 2.157.12 ChartTitle.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartTitle.ToolTip** element specifies the tooltip to display for a [ChartTitle](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **ChartTitle.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **ChartTitle.ToolTip** element.

Parent elements
ChartTitle

The following is the XML Schema definition of the **ChartTitle.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

## 2.158 ChartLegendColumnHeader

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendColumnHeader** element is ignored.

The following are the child elements of the **ChartLegendColumnHeader** element.

Child elements
<a href="#">ChartLegendColumnHeader.Style</a>
<a href="#">ChartLegendColumnHeader.Value</a>

The following is the XML Schema definition of the **ChartLegendColumnHeader** element in RDL 2008/01.

```
<xsd:complexType name="ChartLegendColumnHeaderType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartLegendColumnHeader** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartLegendColumnHeaderType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.158.1 ChartLegendColumnHeader.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendColumnHeader.Style** element is ignored. This element is of type [Style](#).

The following is the parent element of the **ChartLegendColumnHeader.Style** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartLegendColumnHeader</a>
---

The following is the XML Schema definition of the **ChartLegendColumnHeader.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.158.2 ChartLegendColumnHeader.Value

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendColumnHeader.Value** element is ignored.

The following is the parent element of the **ChartLegendColumnHeader.Value** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartLegendColumnHeader</a>
---

The following is the XML Schema definition of the **ChartLegendColumnHeader.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="0" />
```

## 2.159 ChartLegendCustomItems

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItems** element is ignored.

The following is the child element of the **ChartLegendCustomItems** element.

<b>Child elements</b>
-----------------------

<a href="#">ChartLegendCustomItems.ChartLegendCustomItem</a>
--

The following is the XML Schema definition of the **ChartLegendCustomItems** element in RDL 2008/01.

```
<xsd:complexType name="ChartLegendCustomItemsType">
  <xsd:sequence>
    <xsd:element name="ChartLegendCustomItem" type="ChartLegendCustomItemType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartLegendCustomItems** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartLegendCustomItemsType">
  <xsd:sequence>
    <xsd:element name="ChartLegendCustomItem" type="ChartLegendCustomItemType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

```

        maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.159.1 ChartLegendCustomItems.ChartLegendCustomItem

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItems.ChartLegendCustomItem** element is ignored. This element is of type [ChartLegendCustomItem](#).

The following is the parent element of the **ChartLegendCustomItems.ChartLegendCustomItem** element.

Parent elements
<a href="#">ChartLegendCustomItems</a>

The following is the XML Schema definition of the **ChartLegendCustomItems.ChartLegendCustomItem** element.

```

<xsd:element name="ChartLegendCustomItem" type="ChartLegendCustomItemType"
    maxOccurs="unbounded" />

```

### 2.160 ChartLegendCustomItem

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItem** element is ignored.

The following are the attributes and child elements of the **ChartLegendCustomItem** element.

Attributes
<a href="#">ChartLegendCustomItem.Name</a>

Child elements
<a href="#">ChartLegendCustomItem.ActionInfo</a>
<a href="#">ChartLegendCustomItem.ChartLegendCustomItemCells</a>
<a href="#">ChartLegendCustomItem.ChartMarker</a>
<a href="#">ChartLegendCustomItem.Separator</a>
<a href="#">ChartLegendCustomItem.SeparatorColor</a>
<a href="#">ChartLegendCustomItem.Style</a>
<a href="#">ChartLegendCustomItem.ToolTip</a>

The following is the XML Schema definition of the **ChartLegendCustomItem** element in RDL 2008/01.



```

<xsd:complexType name="ChartLegendCustomItemType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartLegendCustomItemCells"
      type="ChartLegendCustomItemCellsType" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
    <xsd:element name="Separator" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **ChartLegendCustomItem** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="ChartLegendCustomItemType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartLegendCustomItemCells"
      type="ChartLegendCustomItemCellsType" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
    <xsd:element name="Separator" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.160.1 ChartLegendCustomItem.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItem.Name** attribute specifies the name for a custom item in a [ChartLegend](#). If the [ChartLegendCustomItem](#) element is specified, the **ChartLegendCustomItem.Name** attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique among the **ChartLegendCustomItem.Name** values in the parent collection.

Following is the parent element of the **ChartLegendCustomItem.Name** attribute.

Parent elements
ChartLegendCustomItem

The following is the XML Schema definition of the **ChartLegendCustomItem.Name** attribute.

```

<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />

```

### 2.160.2 ChartLegendCustomItem.ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItem.ActionInfo** element is ignored. This element is of type [ActionInfo](#).

The following is the parent element of the **ChartLegendCustomItem.ActionInfo** element.

Parent elements
<a href="#">ChartLegendCustomItem</a>

The following is the XML Schema definition of the **ChartLegendCustomItem.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.160.3 ChartLegendCustomItem.ChartLegendCustomItemCells

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendCustomItem.ChartLegendCustomItemCells** element is ignored. This element is of type [ChartLegendCustomItemCells](#).

The following is the parent element of the **ChartLegendCustomItem.ChartLegendCustomItemCells** element.

Parent elements
<a href="#">ChartLegendCustomItemCells</a>

The following is the XML Schema definition of the **ChartLegendCustomItem.ChartLegendCustomItemCells** element.

```
<xsd:element name="ChartLegendCustomItemCells" type="ChartLegendCustomItemCellsType" />
```

### 2.160.4 ChartLegendCustomItem.ChartMarker

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendCustomItem.ChartMarker** element is ignored. This element is of type [ChartMarker](#).

The following is the parent element of the **ChartLegendCustomItem.ChartMarker** element.

Parent elements
<a href="#">ChartLegendCustomItem</a>

The following is the XML Schema definition of the **ChartLegendCustomItem.ChartMarker** element.

```
<xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
```

### 2.160.5 ChartLegendCustomItem.Separator

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendCustomItem.Separator** element is ignored.

The following is the parent element of the **ChartLegendCustomItem.Separator** element.

Parent elements
<a href="#">ChartLegendCustomItem</a>

The following is the XML Schema definition of the **ChartLegendCustomItem.Separator** element.

```
<xsd:element name="Separator" type="xsd:string" minOccurs="0" />
```

### 2.160.6 **ChartLegendCustomItem.SeparatorColor**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItem.SeparatorColor** element is ignored.

The following is the parent element of the **ChartLegendCustomItem.SeparatorColor** element.

Parent elements
<a href="#">ChartLegendCustomItem</a>

The following is the XML Schema definition of the **ChartLegendCustomItem.SeparatorColor** element.

```
<xsd:element name="SeparatorColor" type="xsd:string" minOccurs="0" />
```

### 2.160.7 **ChartLegendCustomItem.Style**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItem.Style** element is ignored. This element is of type [Style](#).

The following is the parent element of the **ChartLegendCustomItem.Style** element.

Parent elements
<a href="#">ChartLegendCustomItem</a>

The following is the XML Schema definition of the **ChartLegendCustomItem.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.160.8 **ChartLegendCustomItem.ToolTip**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItem.ToolTip** element is ignored.

The following is the parent element of the **ChartLegendCustomItem.ToolTip** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartLegendCustomItem</a>
---------------------------------------

The following is the XML Schema definition of the **ChartLegendCustomItem.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

## 2.161 ChartLegendCustomItemCells

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendCustomItemCells** element is ignored.

The following are the parent and child elements of the **ChartLegendCustomItemCells** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartLegendCustomItem</a>
---------------------------------------

<b>Child elements</b>
-----------------------

<a href="#">ChartLegendCustomItemCells.ChartLegendCustomItemCell</a>
--

The following is the XML Schema definition of the **ChartLegendCustomItemCells** element in RDL 2008/01.

```
<xsd:complexType name="ChartLegendCustomItemCellsType">
  <xsd:sequence>
    <xsd:element name="ChartLegendCustomItemCell"
      type="ChartLegendCustomItemCellType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartLegendCustomItemCells** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartLegendCustomItemCellsType">
  <xsd:sequence>
    <xsd:element name="ChartLegendCustomItemCell"
      type="ChartLegendCustomItemCellType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.161.1 ChartLegendCustomItemCells.ChartLegendCustomItemCell

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendCustomItemCells.ChartLegendCustomItemCell** element is ignored. This element is of type [ChartLegendCustomItemCell](#).

The following is the parent element of the **ChartLegendCustomItemCells.ChartLegendCustomItemCell** element.

Parent elements
<a href="#">ChartLegendCustomItemCells</a>

The following is the XML Schema definition of the **ChartLegendCustomItemCells.ChartLegendCustomItemCell** element.

```
<xsd:element name="ChartLegendCustomItemCell" type="ChartLegendCustomItemCellType" maxOccurs="unbounded" />
```

## 2.162 ChartLegendCustomItemCell

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItemCell** element is ignored.

The following are the parent elements, attributes, and child elements of the **ChartLegendCustomItemCell** element.

Parent elements
<a href="#">ChartLegendCustomItemCells</a>

Attributes
<a href="#">ChartLegendCustomItemCell.Name</a>

Child elements
<a href="#">ChartLegendCustomItemCell.ActionInfo</a>
<a href="#">ChartLegendCustomItemCell.Alignment</a>
<a href="#">ChartLegendCustomItemCell.BottomMargin</a>
<a href="#">ChartLegendCustomItemCell.CellSpan</a>
<a href="#">ChartLegendCustomItemCell.CellType</a>
<a href="#">ChartLegendCustomItemCell.ImageHeight</a>
<a href="#">ChartLegendCustomItemCell.ImageWidth</a>
<a href="#">ChartLegendCustomItemCell.LeftMargin</a>
<a href="#">ChartLegendCustomItemCell.RightMargin</a>
<a href="#">ChartLegendCustomItemCell.Style</a>
<a href="#">ChartLegendCustomItemCell.SymbolHeight</a>
<a href="#">ChartLegendCustomItemCell.SymbolWidth</a>

Child elements
<a href="#">ChartLegendCustomItemCell.Text</a>
<a href="#">ChartLegendCustomItemCell.ToolTip</a>
<a href="#">ChartLegendCustomItemCell.TopMargin</a>

The following is the XML Schema definition of the **ChartLegendCustomItemCell** element in RDL 2008/01.

```
<xsd:complexType name="ChartLegendCustomItemCellType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="CellType" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Text" />
          <xsd:enumeration value="SeriesSymbol" />
          <xsd:enumeration value="Image" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="CellSpan" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ImageHeight" type="xsd:string" minOccurs="0" />
    <xsd:element name="ImageWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="SymbolHeight" type="xsd:string" minOccurs="0" />
    <xsd:element name="SymbolWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="Alignment" type="xsd:string" minOccurs="0" />
    <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ChartLegendCustomItemCell** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ChartLegendCustomItemCellType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="CellType" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Text" />
          <xsd:enumeration value="SeriesSymbol" />
          <xsd:enumeration value="Image" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="CellSpan" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ImageHeight" type="xsd:string" minOccurs="0" />
    <xsd:element name="ImageWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="SymbolHeight" type="xsd:string" minOccurs="0" />
    <xsd:element name="SymbolWidth" type="xsd:string" minOccurs="0" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

```

<xsd:element name="Alignment" type="xsd:string" minOccurs="0" />
<xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.162.1 ChartLegendCustomItemCell.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItemCell.Name** attribute specifies the name of the cell of a custom item in a [ChartLegend](#). If the [ChartLegendCustomItemCell](#) element is specified, the **ChartLegendCustomItemCell.Name** attribute MUST be specified, and the value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [UTR15] that is unique among the **ChartLegendCustomItemCell.Name** values in the parent collection.

The following is the parent element of the **ChartLegendCustomItemCell.Name** attribute.

Parent elements
<a href="#">ChartLegendCustomItemCell</a>

The following is the XML Schema definition of the **ChartLegendCustomItemCell.Name** attribute.

```

<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />

```

### 2.162.2 ChartLegendCustomItemCell.ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItemCell.ActionInfo** element is ignored.

The following is the parent element of the **ChartLegendCustomItemCell.ActionInfo** element.

Parent elements
<a href="#">ChartLegendCustomItemCell</a>

The following is the XML Schema definition of the **ChartLegendCustomItemCell.ActionInfo** element.

```

<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />

```

### 2.162.3 ChartLegendCustomItemCell.Alignment

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItemCell.Alignment** element is ignored.

The following is the parent element of the **ChartLegendCustomItemCell.Alignment** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartLegendCustomItemCell</a>
---

The following is the XML Schema definition of the **ChartLegendCustomItemCell.Alignment** element.

```
<xsd:element name="Alignment" type="xsd:string" minOccurs="0" />
```

## 2.162.4 ChartLegendCustomItemCell.BottomMargin

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendCustomItemCell.BottomMargin** element is ignored.

The following is the parent element of the **ChartLegendCustomItemCell.BottomMargin** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartLegendCustomItemCell</a>
---

The following is the XML Schema definition of the **ChartLegendCustomItemCell.BottomMargin** element.

```
<xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
```

## 2.162.5 ChartLegendCustomItemCell.CellSpan

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendCustomItemCell.CellSpan** element is ignored.

The following is the parent element of the **ChartLegendCustomItemCell.CellSpan** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartLegendCustomItemCell</a>
---

The following is the XML Schema definition of the **ChartLegendCustomItemCell.CellSpan** element.

```
<xsd:element name="CellSpan" type="xsd:unsignedInt" minOccurs="0" />
```

## 2.162.6 ChartLegendCustomItemCell.CellType

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendCustomItemCell.CellType** element is ignored.

The following is the parent element of the **ChartLegendCustomItemCell.CellType** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartLegendCustomItemCell</a>
---



The following is the XML Schema definition of the **ChartLegendCustomItemCell.CellType** element.

```
<xsd:element name="CellType" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Text" />
      <xsd:enumeration value="SeriesSymbol" />
      <xsd:enumeration value="Image" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.162.7 ChartLegendCustomItemCell.ImageHeight

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItemCell.ImageHeight** element is ignored.

The following is the parent element of the **ChartLegendCustomItemCell.ImageHeight** element.

Parent elements
<a href="#">ChartLegendCustomItemCell</a>

The following is the XML Schema definition of the **ChartLegendCustomItemCell.ImageHeight** element.

```
<xsd:element name="ImageHeight" type="xsd:string" minOccurs="0" />
```

## 2.162.8 ChartLegendCustomItemCell.ImageWidth

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItemCell.ImageWidth** element is ignored.

The following is the parent element of the **ChartLegendCustomItemCell.ImageWidth** element.

Parent elements
<a href="#">ChartLegendCustomItemCell</a>

The following is the XML Schema definition of the **ChartLegendCustomItemCell.ImageWidth** element.

```
<xsd:element name="ImageWidth" type="xsd:string" minOccurs="0" />
```

## 2.162.9 ChartLegendCustomItemCell.LeftMargin

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItemCell.LeftMargin** element is ignored.

The following is the parent element of the **ChartLegendCustomItemCell.LeftMargin** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartLegendCustomItemCell</a>
---

The following is the XML Schema definition of the **ChartLegendCustomItemCell.LeftMargin** element.

```
<xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
```

## 2.162.10 ChartLegendCustomItemCell.RightMargin

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendCustomItemCell.RightMargin** element is ignored.

The following is the parent element of the **ChartLegendCustomItemCell.RightMargin** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartLegendCustomItemCell</a>
---

The following is the XML Schema definition of the **ChartLegendCustomItemCell.RightMargin** element.

```
<xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
```

## 2.162.11 ChartLegendCustomItemCell.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendCustomItemCell.Style** element is ignored. This element is of type [Style](#).

The following is the parent element of the **ChartLegendCustomItemCell.Style** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartLegendCustomItemCell</a>
---

The following is the XML Schema definition of the **ChartLegendCustomItemCell.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.162.12 ChartLegendCustomItemCell.SymbolHeight

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendCustomItemCell.SymbolHeight** element is ignored.

The following is the parent element of the **ChartLegendCustomItemCell.SymbolHeight** element.

<b>Parent elements</b>
------------------------

<a href="#">ChartLegendCustomItemCell</a>
---

The following is the XML Schema definition of the **ChartLegendCustomItemCell.SymbolHeight** element.

```
<xsd:element name="SymbolHeight" type="xsd:string" minOccurs="0" />
```

### 2.162.13 ChartLegendCustomItemCell.SymbolWidth

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendCustomItemCell.SymbolWidth** element is ignored.

The following is the parent element of the **ChartLegendCustomItemCell.SymbolWidth** element.

Parent elements
<a href="#">ChartLegendCustomItemCell</a>

The following is the XML Schema definition of the **ChartLegendCustomItemCell.SymbolWidth** element.

```
<xsd:element name="SymbolWidth" type="xsd:string" minOccurs="0" />
```

### 2.162.14 ChartLegendCustomItemCell.Text

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendCustomItemCell.Text** element is ignored.

The following is the parent element of the **ChartLegendCustomItemCell.Text** element.

Parent elements
<a href="#">ChartLegendCustomItemCell</a>

The following is the XML Schema definition of the **ChartLegendCustomItemCell.Text** element.

```
<xsd:element name="Text" type="xsd:string" minOccurs="0" />
```

### 2.162.15 ChartLegendCustomItemCell.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ChartLegendCustomItemCell.ToolTip** element is ignored.

The following is the parent element of the **ChartLegendCustomItemCell.ToolTip** element.

Parent elements
<a href="#">ChartLegendCustomItemCell</a>

The following is the XML Schema definition of the **ChartLegendCustomItemCell.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

## 2.162.16 ChartLegendCustomItemCell.TopMargin

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ChartLegendCustomItemCell.TopMargin** element is ignored.

The following is the parent element of the **ChartLegendCustomItemCell.TopMargin** element.

Parent elements
<a href="#">ChartLegendCustomItemCell</a>

The following is the XML Schema definition of the **ChartLegendCustomItemCell.TopMargin** element.

```
<xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
```

## 2.163 GaugePanel

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel** element specifies **gauge** visualization for a data point or a set of data points.

The following are the parent elements, attributes, and child elements of the **GaugePanel** element.

Parent elements
<a href="#">ReportItems</a>
<a href="#">CellContents</a>
<a href="#">CustomReportItem.AltReportItem</a>

Attributes
<a href="#">GaugePanel.Name</a>

Child elements
<a href="#">GaugePanel.Style</a>
<a href="#">GaugePanel.ActionInfo</a>
<a href="#">GaugePanel.Bookmark</a>
<a href="#">GaugePanel.CustomProperties</a>
<a href="#">GaugePanel.DataElementName</a>
<a href="#">GaugePanel.DataElementOutput</a>
<a href="#">GaugePanel.DocumentMapLabel</a>
<a href="#">GaugePanel.Height</a>

Child elements
<a href="#">GaugePanel.Left</a>
<a href="#">GaugePanel.RepeatWith</a>
<a href="#">GaugePanel.ToolTip</a>
<a href="#">GaugePanel.Top</a>
<a href="#">GaugePanel.Visibility</a>
<a href="#">GaugePanel.Width</a>
<a href="#">GaugePanel.ZIndex</a>
<a href="#">GaugePanel.DataSetName</a>
<a href="#">GaugePanel.Filters</a>
<a href="#">GaugePanel.NoRowsMessage</a>
<a href="#">GaugePanel.PageBreak</a>
<a href="#">GaugePanel.PageName</a>
<a href="#">GaugePanel.SortExpressions</a>
<a href="#">GaugePanel.AntiAliasing</a>
<a href="#">GaugePanel.AutoLayout</a>
<a href="#">GaugePanel.BackFrame</a>
<a href="#">GaugePanel.GaugeImages</a>
<a href="#">GaugePanel.GaugeLabels</a>
<a href="#">GaugePanel.GaugeMember</a>
<a href="#">GaugePanel.LinearGauges</a>
<a href="#">GaugePanel.NumericIndicators</a>
<a href="#">GaugePanel.RadialGauges</a>
<a href="#">GaugePanel.ShadowIntensity</a>
<a href="#">GaugePanel.StateIndicators</a>
<a href="#">GaugePanel.TextAntiAliasingQuality</a>
<a href="#">GaugePanel.TopImage</a>

**Applies to [RDL 2011/01](#)**

Child elements
<a href="#">GaugePanel.Relationship</a>

The following is the XML Schema definition of the **GaugePanel** element in RDL 2008/01.

```
<xsd:complexType name="GaugePanelType">
```

```

<xsd:choice minOccurs="0" maxOccurs="unbounded">
  <!--DataRegionTypeStart-->
  <xsd:element name="Style" type="StyleType" minOccurs="0" />
  <xsd:element name="SortExpressions" type="SortExpressionsType"
    minOccurs="0" />
  <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
  <xsd:element name="Top" type="SizeType" minOccurs="0" />
  <xsd:element name="Left" type="SizeType" minOccurs="0" />
  <xsd:element name="Height" type="SizeType" minOccurs="0" />
  <xsd:element name="Width" type="SizeType" minOccurs="0" />
  <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
  <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
  <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
  <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
  <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
  <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
  <xsd:element name="CustomProperties" type="CustomPropertiesType"
    minOccurs="0" />
  <xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
  <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
  <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
  <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
  <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
  <xsd:element name="DataElementOutput" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Output" />
        <xsd:enumeration value="NoOutput" />
        <xsd:enumeration value="ContentsOnly" />
        <xsd:enumeration value="Auto" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <!--DataRegionTypeEnd-->
  <xsd:element name="AntiAliasing" type="xsd:string" minOccurs="0" />
  <xsd:element name="TextAntiAliasingQuality" type="xsd:string" minOccurs="0" />
  <xsd:element name="AutoLayout" type="xsd:string" minOccurs="0" />
  <xsd:element name="ShadowIntensity" type="xsd:string" minOccurs="0" />
  <xsd:element name="RadialGauges" type="RadialGaugesType" minOccurs="0" />
  <xsd:element name="LinearGauges" type="LinearGaugesType" minOccurs="0" />
  <xsd:element name="NumericIndicators" type="NumericIndicatorsType"
    minOccurs="0" />
  <xsd:element name="StateIndicators" type="StateIndicatorsType"
    minOccurs="0" />
  <xsd:element name="GaugeImages" type="GaugeImagesType" minOccurs="0" />
  <xsd:element name="GaugeLabels" type="GaugeLabelsType" minOccurs="0" />
  <xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
  <xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
  <xsd:element name="GaugeMember" type="GaugeMemberType" minOccurs="0" />
  <xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **GaugePanel** element in RDL 2010/01 and RDL 2016/01.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```

<xsd:complexType name="GaugePanelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--DataRegionTypeStart-->

```

```

<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="SortExpressions" type="SortExpressionsType"
minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="Top" type="SizeType" minOccurs="0" />
<xsd:element name="Left" type="SizeType" minOccurs="0" />
<xsd:element name="Height" type="SizeType" minOccurs="0" />
<xsd:element name="Width" type="SizeType" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType"
minOccurs="0" />
<xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
<xsd:element name="PageName" type="xsd:string" minOccurs="0" />
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<!--DataRegionTypeEnd-->
<xsd:element name="AntiAliasing" type="xsd:string" minOccurs="0" />
<xsd:element name="TextAntiAliasingQuality" type="xsd:string" minOccurs="0" />
<xsd:element name="AutoLayout" type="xsd:string" minOccurs="0" />
<xsd:element name="ShadowIntensity" type="xsd:string" minOccurs="0" />
<xsd:element name="RadialGauges" type="RadialGaugesType" minOccurs="0" />
<xsd:element name="LinearGauges" type="LinearGaugesType" minOccurs="0" />
<xsd:element name="NumericIndicators" type="NumericIndicatorsType"
minOccurs="0" />
<xsd:element name="StateIndicators" type="StateIndicatorsType"
minOccurs="0" />
<xsd:element name="GaugeImages" type="GaugeImagesType" minOccurs="0" />
<xsd:element name="GaugeLabels" type="GaugeLabelsType" minOccurs="0" />
<xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
<xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
<xsd:element name="GaugeMember" type="GaugeMemberType" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.163.1 GaugePanel.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.Name** attribute specifies a unique identifier for a [GaugePanel](#). The **GaugePanel.Name** attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [UTR15] that is unique for all the report items within a [Report](#).

The following is the parent element of the **GaugePanel.Name** attribute.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.163.2 GaugePanel.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.Style** element specifies style information for a [GaugePanel](#). The **GaugePanel.Style** element is optional. This element is of type [Style](#).

The following is the parent element of the **GaugePanel.Style** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0">
```

### 2.163.3 GaugePanel.ActionInfo

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.ActionInfo** element is ignored.

The following is the parent element of the **GaugePanel.ActionInfo** element.

Parent elements
<a href="#">GaugePanel</a>

The following is the XML Schema definition of the **GaugePanel.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0">
```

### 2.163.4 GaugePanel.Bookmark

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.Bookmark** element specifies a bookmark that can be linked to via a bookmark action for a [GaugePanel](#). The **GaugePanel.Bookmark** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **GaugePanel.Bookmark** element.



Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.Bookmark** element.

```
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0">
```

### 2.163.5 GaugePanel.CustomProperties

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.CustomProperties** element specifies custom information for a [GaugePanel](#) that will be handed to a report rendering component. The **GaugePanel.CustomProperties** element is optional. This element is of type [CustomProperties](#).

The following is the parent element of the **GaugePanel.CustomProperties** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

### 2.163.6 GaugePanel.DataElementName

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.DataElementName** element specifies the name of a [GaugePanel](#) to be used in a **data rendering**. The **GaugePanel.DataElementName** element is optional.

If this element is not present, its value is interpreted to be the **Name** attribute of the **gauge panel**. If the **GaugePanel.DataElementName** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String** that is a **CLS-compliant identifier** [\[UTR15\]](#).

The following is the parent element of the **GaugePanel.DataElementName** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0">
```

### 2.163.7 GaugePanel.DataElementOutput

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.DataElementOutput** element specifies whether a [GaugePanel](#) appears in a data (XML, CSV) rendering. The **GaugePanel.DataElementOutput** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is one of the following:

**Auto:** Specifies the default setting for how the **gauge panel** will appear in a **data rendering**, which is the same as "NoOutput" if [Visibility.Hidden](#) for the gauge panel is set to true and [Visibility.ToggleItem](#) is not set. Otherwise, this value is the same as "Output".

**Output:** Specifies that the gauge panel appears in a data rendering output.

**NoOutput:** Specifies that the gauge panel does not appear in a data rendering output.

If the **GaugePanel.DataElementOutput** element is not present, its value is interpreted as "Auto".

The following is the parent element of the **GaugePanel.DataElementOutput** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.163.8 GaugePanel.DocumentMapLabel

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.DocumentMapLabel** element specifies a label to identify a [GaugePanel](#) within the client UI in order to provide a user-friendly label for searching. The **GaugePanel.DocumentMapLabel** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. This element MUST NOT be contained within a [PageSection](#).

The following is the parent element of the **GaugePanel.DocumentMapLabel** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.DocumentMapLabel** element.

```
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
```

### 2.163.9 GaugePanel.Height

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.Height** element specifies the height of a [GaugePanel](#). The **GaugePanel.Height** element is optional. If this element is present, its value MUST be an [RdlSize](#). If the **GaugePanel.Height** element is not present, its value is interpreted as the height of the **gauge panel**'s container minus the value of the peer [GaugePanel.Top](#) element.

The following is the parent element of the **GaugePanel.Height** element.

Parent elements
<a href="#">GaugePanel</a>

The following is the XML Schema definition of the **GaugePanel.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="0" />
```

### 2.163.10 GaugePanel.Left

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.Left** element specifies the distance of a [GaugePanel](#) from the left of the **gauge panel**'s container. The **GaugePanel.Left** element is optional. If this element is present, its value MUST be an [RdlSize](#).

The following is the parent element of the **GaugePanel.Left** element.

Parent elements
<a href="#">GaugePanel</a>

The following is the XML Schema definition of the **GaugePanel.Left** element.

```
<xsd:element name="Left" type="SizeType" minOccurs="0">
```

### 2.163.11 GaugePanel.RepeatWith

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.RepeatWith** element MUST NOT be specified.

The following is the parent element of the **GaugePanel.RepeatWith** element.

Parent elements
<a href="#">GaugePanel</a>

The following is the XML Schema definition of the **GaugePanel.RepeatWith** element.

```
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0">
```

### 2.163.12 GaugePanel.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.ToolTip** element specifies a textual label for a [GaugePanel](#) that is used for things such as specifying **title** and **alt** attributes in HTML reports. The **alt** attribute can be used to render alternative text (alt text). The **GaugePanel.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **GaugePanel.ToolTip** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0">
```

### 2.163.13 GaugePanel.Top

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.Top** element specifies the distance of a [GaugePanel](#) from the top of **gauge panel**'s container. The **GaugePanel.Top** element is optional. If this element is present, its value MUST be an [RdlSize](#).

The following is the parent element of the **GaugePanel.Top** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.Top** element.

```
<xsd:element name="Top" type="SizeType" minOccurs="0">
```

### 2.163.14 GaugePanel.Visibility

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.Visibility** element specifies whether a [GaugePanel](#) is hidden. The **GaugePanel.Visibility** element is optional. This element is of type [Visibility](#).

The following is the parent element of the **GaugePanel.Visibility** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0">
```

### 2.163.15 GaugePanel.Width

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.Width** element specifies the width of a [GaugePanel](#). The **GaugePanel.Width** element is optional. If this element is present, its value MUST be an [RdlSize](#). If the **GaugePanel.Width** element is not present, its value is interpreted as the width of the **gauge panel's** container (such as a [Rectangle](#) or [Body](#)) minus the value of the peer [GaugePanel.Left](#) element, if specified.

The following is the parent element of the **GaugePanel.Width** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="0">
```

### 2.163.16 GaugePanel.ZIndex

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.ZIndex** element specifies the drawing order of a [GaugePanel](#) within its container. The **GaugePanel.ZIndex** element is optional. If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17). The value of this element MUST greater than or equal to 0 and less than or equal to 2147483647. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **GaugePanel.ZIndex** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0">
```

### 2.163.17 GaugePanel.DataSetName

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.DataSetName** element specifies the name of the [DataSet](#) to use to bind data to a [GaugePanel](#). The **GaugePanel.DataSetName** element MUST be present if there is not exactly one and only one [DataSets.DataSet](#) instance contained within the associated [Report](#). If there is only one **Datasets.Dataset** element in the entire report, the **GaugePanel.DataSetName** element is optional; that dataset will be used. The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

If the **GaugePanel** has an ancestor, the value of the **GaugePanel.DataSetName** element is interpreted as the **DataSet.Name** for the containing scope (**DataRegion**, [Group](#), or **Cell**).[<33>](#)

The following is the parent element of the **GaugePanel.DataSetName** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0">
```

### 2.163.18 GaugePanel.Filters

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.Filters** element specifies filters to apply to each row of data associated within a [GaugePanel](#). The **GaugePanel.Filters** element is optional. This element is of type [Filters](#).

The following is the parent element of the **GaugePanel.Filters** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

### 2.163.19 GaugePanel.NoRowsMessage

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.NoRowsMessage** element specifies a message to display within a [GaugePanel](#) when no rows of data are associated with the gauge panel. The **GaugePanel.NoRowsMessage** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The information from [GaugePanel.Style](#), if present, MUST be applied to the text of the **GaugePanel.NoRowsMessage** element.

The following is the parent element of the **GaugePanel.NoRowsMessage** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.NoRowsMessage** element.

```
<xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0">
```

### 2.163.20 GaugePanel.PageBreak

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.PageBreak** element specifies page break behavior for a [GaugePanel](#). The **GaugePanel.PageBreak** element is optional. This element is of type [PageBreak](#).

The following is the parent element of the **GaugePanel.PageBreak** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.PageBreak** element.

```
<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0">
```

### 2.163.21 GaugePanel.PageName

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.PageName** element specifies the value to use for the name of a paginated page. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **GaugePanel.PageName** element.

Parent elements
<a href="#">GaugePanel</a>

The following is the XML Schema definition of the **GaugePanel.PageName** element.

```
<xsd:element name="PageName" type="xsd:string" minOccurs="0" />
```

### 2.163.22 GaugePanel.SortExpressions

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.SortExpressions** element specifies the expressions by which to sort the rows of data associated with a [GaugePanel](#). The **GaugePanel.SortExpressions** element is optional. This element is of type [SortExpressions](#).

The following is the parent element of the **GaugePanel.SortExpressions** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.SortExpressions** element.

```
<xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
```

### 2.163.23 GaugePanel.AntiAliasing

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.AntiAliasing** element specifies the **anti-aliasing** type for a [GaugePanel](#). The **GaugePanel.AntiAliasing** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The **String** value of this element MUST be one of the following:

**All:** Both text and graphics have anti-aliasing applied.

**Text:** Only text has anti-aliasing applied.

**Graphics:** Graphical elements have anti-aliasing applied.

**None:** Nothing in the gauge panel has anti-aliasing applied.

If the **GaugePanel.AntiAliasing** element is not present, its **String** value is interpreted as "All".

The following is the parent element of the **GaugePanel.AntiAliasing** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.AntiAliasing** element.

```
<xsd:element name="AntiAliasing" type="xsd:string" minOccurs="0">
```

### 2.163.24 GaugePanel.AutoLayout

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.AutoLayout** element specifies whether automatic layout is used for elements in a [GaugePanel](#). The **GaugePanel.AutoLayout** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **GaugePanel.AutoLayout** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.AutoLayout** element.

```
<xsd:element name="AutoLayout" type="xsd:string" minOccurs="0">
```

### 2.163.25 GaugePanel.BackFrame

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.BackFrame** element specifies the properties for a frame and frame background for a [GaugePanel](#). The **GaugePanel.BackFrame** element is optional. This element is of type [BackFrame](#).

The following is the parent element of the **GaugePanel.BackFrame** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.BackFrame** element.

```
<xsd:element name="BackFrame" type="BackFrameType" minOccurs="0">
```



## 2.163.26 GaugePanel.GaugeImages

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.GaugeImages** element is of type [GaugeImages](#). The **GaugePanel.GaugeImages** element is ignored.

The following is the parent element of the **GaugePanel.GaugeImages** element.

Parent elements
<a href="#">GaugePanel</a>

The following is the XML Schema definition of the **GaugePanel.GaugeImages** element.

```
<xsd:element name="GaugeImages" type="GaugeImagesType" minOccurs="0" />
```

## 2.163.27 GaugePanel.GaugeLabels

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.GaugeLabels** element specifies the set of **gauge** labels in a [GaugePanel](#). The **GaugePanel.GaugeLabels** element is optional. This element is of type [GaugeLabels](#).

The following is the parent element of the **GaugePanel.GaugeLabels** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.GaugeLabels** element.

```
<xsd:element name="GaugeLabels" type="GaugeLabelsType" minOccurs="0">
```

## 2.163.28 GaugePanel.GaugeMember

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.GaugeMember** element specifies group, sort, and filter behavior for the data in a [GaugePanel](#). The **GaugePanel.GaugeMember** element is optional. This element is of type [GaugeMember](#).

The following is the parent element of the **GaugePanel.GaugeMember** element.

Parent elements
GaugePanel

The following is the XML Schema definition of the **GaugePanel.GaugeMember** element.

```
<xsd:element name="GaugeMember" type="GaugeMemberType" minOccurs="0">
```

## 2.163.29 GaugePanel.LinearGauges

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.LinearGauges** element specifies the set of [LinearGauge](#) instances in a [GaugePanel](#). The **GaugePanel.LinearGauges** element is optional. This element is of type [LinearGauges](#).

The following is the parent element of the **GaugePanel.LinearGauges** element.

Parent elements
<a href="#">GaugePanel</a>

The following is the XML Schema definition of the **GaugePanel.LinearGauges** element.

```
<xsd:element name="LinearGauges" type="LinearGaugesType" minOccurs="0">
```

## 2.163.30 GaugePanel.NumericIndicators

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.NumericIndicators** element is of type [NumericIndicators](#). This element is ignored. However, the type of the **GaugePanel.NumericIndicators** element is validated.

The following is the parent element of the **GaugePanel.NumericIndicators** element.

Parent elements
<a href="#">GaugePanel</a>

The following is the XML Schema definition of the **GaugePanel.NumericIndicators** element.

```
<xsd:element name="NumericIndicators" type="NumericIndicatorsType" minOccurs="0" />
```

## 2.163.31 GaugePanel.RadialGauges

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.RadialGauges** element specifies the set of [RadialGauge](#) instances in a [GaugePanel](#). The **GaugePanel.RadialGauges** element is optional. This element is of type [RadialGauges](#).

The following is the parent element of the **GaugePanel.RadialGauges** element.

Parent elements
<a href="#">GaugePanel</a>

The following is the XML Schema definition of the **GaugePanel.RadialGauges** element.

```
<xsd:element name="RadialGauges" type="RadialGaugesType" minOccurs="0">
```

## 2.163.32 GaugePanel.ShadowIntensity

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.ShadowIntensity** element specifies the intensity of shadows throughout a [GaugePanel](#). The **GaugePanel.ShadowIntensity** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to **Float**. The **Float** value MUST be greater than or equal to 0 and less than or equal to 100 (as a percentage of maximum intensity), with 0 corresponding to the lowest shadow intensity, and 100 corresponding to the highest shadow intensity. If this element is not present, its value is interpreted as 25.

The following is the parent element of the **GaugePanel.ShadowIntensity** element.

Parent elements
<a href="#">GaugePanel</a>

The following is the XML Schema definition of the **GaugePanel.ShadowIntensity** element.

```
<xsd:element name="ShadowIntensity" type="xsd:string" minOccurs="0" />
```

## 2.163.33 GaugePanel.StateIndicators

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.StateIndicators** element is of type [StateIndicators](#). The **GaugePanel.StateIndicators** element is ignored in any schema version prior to RDL 2010/01.

The following is the parent element of the **GaugePanel.StateIndicators** element.

Parent elements
<a href="#">GaugePanel</a>

The following is the XML Schema definition of the **GaugePanel.StateIndicators** element.

```
<xsd:element name="StateIndicators" type="StateIndicatorsType" minOccurs="0" />
```

## 2.163.34 GaugePanel.TextAntiAliasingQuality

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugePanel.TextAntiAliasingQuality** element specifies the **anti-aliasing** quality for text in a [GaugePanel](#). The **GaugePanel.TextAntiAliasingQuality** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The specified value for this element MUST always be set to one of the following:

**High:** The text in the gauge panel will have a high anti-aliasing quality.

**Normal:** The text in the gauge panel will have a normal anti-aliasing quality.

**SystemDefault:** The text in the gauge panel will have an anti-aliasing quality, which is the system default.

If the **GaugePanel.TextAntiAliasingQuality** element is not present, its value is interpreted as "High".

The following is the parent element of the **GaugePanel.TextAntiAliasingQuality** element.

Parent elements
<a href="#">GaugePanel</a>

The following is the XML Schema definition of the **GaugePanel.TextAntiAliasingQuality** element.

```
<xsd:element name="TextAntiAliasingQuality" type="xsd:string" minOccurs="0" />
```

### 2.163.35 GaugePanel.TopImage

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugePanel.TopImage** element specifies an image to be displayed at the top part of a **gauge**. The **GaugePanel.TopImage** element is optional. This element is of type [TopImage](#).

The following is the parent element of the **GaugePanel.TopImage** element.

Parent elements
<a href="#">GaugePanel</a>

The following is the XML Schema definition of the **GaugePanel.TopImage** element.

```
<xsd:element name="TopImage" type="TopImageType" minOccurs="0">
```

### 2.163.36 GaugePanel.Relationship

*Applies to [RDL 2011/01](#)*

The **GaugePanel.Relationship** element specifies a **relationship** to use for correlating data in a [GaugePanel](#) with the data in the containing scope. The **GaugePanel.Relationship** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [Relationship](#). The **GaugePanel.Relationship** element is ignored if the dataset for this **GaugePanel** is the same as the dataset for each containing scope. The **GaugePanel.Relationship** element MUST NOT be specified if there is no containing scope.

Following is the parent element of the **GaugePanel.Relationship** element.

Parent elements
<a href="#">GaugePanel</a>

The following is the XML Schema definition of the **GaugePanel.Relationship** element.

```
<xsd:element name="Relationship" type="RelationshipType" minOccurs="0" />
```

## 2.164 BackFrame

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **BackFrame** element specifies the frame and frame background properties for a [LinearGauge](#), a [RadialGauge](#), or a [GaugePanel](#). This element is optional.

The following are the parent and child elements of the **BackFrame** element.

Parent elements
<a href="#">LinearGauge</a>
<a href="#">RadialGauge</a>
<a href="#">GaugePanel</a>

Child elements
<a href="#">BackFrame.FrameBackground</a>
<a href="#">BackFrame.FrameImage</a>
<a href="#">BackFrame.FrameShape</a>
<a href="#">BackFrame.FrameStyle</a>
<a href="#">BackFrame.FrameWidth</a>
<a href="#">BackFrame.GlassEffect</a>
<a href="#">BackFrame.Style</a>

The following is the XML Schema definition of the **BackFrame** element in RDL 2008/01.

```
<xsd:complexType name="BackFrameType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="FrameBackground" type="FrameBackgroundType"
      minOccurs="0" />
    <xsd:element name="FrameImage" type="FrameImageType" minOccurs="0" />
    <xsd:element name="FrameStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="FrameShape" type="xsd:string" minOccurs="0" />
    <xsd:element name="FrameWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="GlassEffect" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **BackFrame** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="BackFrameType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="FrameBackground" type="FrameBackgroundType"
      minOccurs="0" />
    <xsd:element name="FrameImage" type="FrameImageType" minOccurs="0" />
    <xsd:element name="FrameStyle" type="xsd:string" minOccurs="0" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

```

<xsd:element name="FrameShape" type="xsd:string" minOccurs="0" />
<xsd:element name="FrameWidth" type="xsd:string" minOccurs="0" />
<xsd:element name="GlassEffect" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.164.1 BackFrame.FrameBackground

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **BackFrame.FrameBackground** element specifies the appearance properties of a [BackFrame](#). The **BackFrame.FrameBackground** element is optional. This element is of type [FrameBackground](#).

The following is the parent element of the **BackFrame.FrameBackground** element.

Parent elements
BackFrame

The following is the XML Schema definition of the **BackFrame.FrameBackground** element.

```
<xsd:element name="FrameBackground" type="FrameBackgroundType" minOccurs="0" />
```

### 2.164.2 BackFrame.FrameImage

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **BackFrame.FrameImage** element specifies image properties for a [BackFrame](#). The **BackFrame.FrameImage** element is optional. This element is of type [FrameImage](#).

The following is the parent element of the **BackFrame.FrameImage** element.

Parent elements
BackFrame

The following is the XML Schema definition of the **BackFrame.FrameImage** element.

```
<xsd:element name="FrameImage" type="FrameImageType" minOccurs="0" />
```

### 2.164.3 BackFrame.FrameShape

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **BackFrame.FrameShape** element specifies the shape of a [BackFrame](#). The **BackFrame.FrameShape** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies that this value is interpreted as "Circular" for [RadialGauge](#) instances and "Rectangular" for [LinearGauge](#) and [GaugePanel](#) instances.

**Circular:** Specifies that the shape of the frame is a circle.

**Rectangular:** Specifies that the shape of the frame is a **rectangle**.

**RoundedRectangular:** Specifies that the shape of the frame is a rounded rectangle.

**AutoShape:** Specifies that the shape of the frame is automatically determined in such a way that it would encompass all the elements of the **gauge** or the gauge container.

**CustomCircular1:** Specifies a custom circular shape.

**CustomCircular2:** Specifies a custom circular shape.

**CustomCircular3:** Specifies a custom circular shape.

**CustomCircular4:** Specifies a custom circular shape.

**CustomCircular5:** Specifies a custom circular shape.

**CustomCircular6:** Specifies a custom circular shape.

**CustomCircular7:** Specifies a custom circular shape.

**CustomCircular8:** Specifies a custom circular shape.

**CustomCircular9:** Specifies a custom circular shape.

**CustomCircular10:** Specifies a custom circular shape.

**CustomCircular11:** Specifies a custom circular shape.

**CustomCircular12:** Specifies a custom circular shape.

**CustomCircular13:** Specifies a custom circular shape.

**CustomCircular14:** Specifies a custom circular shape.

**CustomCircular15:** Specifies a custom circular shape.

**CustomSemiCircularN1:** Specifies a custom semi-circular shape that faces north.

**CustomSemiCircularN2:** Specifies a custom semi-circular shape that faces north.

**CustomSemiCircularN3:** Specifies a custom semi-circular shape that faces north.

**CustomSemiCircularN4:** Specifies a custom semi-circular shape that faces north.

**CustomSemiCircularS1:** Specifies a custom semi-circular shape that faces south.

**CustomSemiCircularS2:** Specifies a custom semi-circular shape that faces south.

**CustomSemiCircularS3:** Specifies a custom semi-circular shape that faces south.

**CustomSemiCircularS4:** Specifies a custom semi-circular shape that faces south.

**CustomSemiCircularE1:** Specifies a custom semi-circular shape that faces east.

**CustomSemiCircularE2:** Specifies a custom semi-circular shape that faces east.

**CustomSemiCircularE3:** Specifies a custom semi-circular shape that faces east.

**CustomSemiCircularE4:** Specifies a custom semi-circular shape that faces east.

**CustomSemiCircularW1:** Specifies a custom semi-circular shape that faces west.

**CustomSemiCircularW2**: Specifies a custom semi-circular shape that faces west.

**CustomSemiCircularW3**: Specifies a custom semi-circular shape that faces west.

**CustomSemiCircularW4**: Specifies a custom semi-circular shape that faces west.

**CustomQuarterCircularNE1**: Specifies a custom quarter-circular shape that faces northeast.

**CustomQuarterCircularNE2**: Specifies a custom quarter-circular shape that faces northeast.

**CustomQuarterCircularNE3**: Specifies a custom quarter-circular shape that faces northeast.

**CustomQuarterCircularNE4**: Specifies a custom quarter-circular shape that faces northeast.

**CustomQuarterCircularNW1**: Specifies a custom quarter-circular shape that faces northwest.

**CustomQuarterCircularNW2**: Specifies a custom quarter-circular shape that faces northwest.

**CustomQuarterCircularNW3**: Specifies a custom quarter-circular shape that faces northwest.

**CustomQuarterCircularNW4**: Specifies a custom quarter-circular shape that faces northwest.

**CustomQuarterCircularSE1**: Specifies a custom quarter-circular shape that faces southeast.

**CustomQuarterCircularSE2**: Specifies a custom quarter-circular shape that faces southeast.

**CustomQuarterCircularSE3**: Specifies a custom quarter-circular shape that faces southeast.

**CustomQuarterCircularSE4**: Specifies a custom quarter-circular shape that faces southeast.

**CustomQuarterCircularSW1**: Specifies a custom quarter-circular shape that faces southwest.

**CustomQuarterCircularSW2**: Specifies a custom quarter-circular shape that faces southwest.

**CustomQuarterCircularSW3**: Specifies a custom quarter-circular shape that faces southwest.

**CustomQuarterCircularSW4**: Specifies a custom quarter-circular shape that faces southwest.

If the **BackFrame.FrameShape** element is not present, its value is interpreted as "Default".

The following is the parent element of the **BackFrame.FrameShape** element.

Parent elements
BackFrame

The following is the XML Schema definition of the **BackFrame.FrameShape** element.

```
<xsd:element name="FrameShape" type="xsd:string" minOccurs="0" />
```

## 2.164.4 BackFrame.FrameStyle

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **BackFrame.FrameStyle** element specifies the general style appearance for a [BackFrame](#) instance. The **BackFrame.FrameStyle** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**None**: Specifies that there is no frame.



**Simple:** Specifies that there is a very simple frame.

**Edged:** Specifies that there is a frame with a single edge.

If this element is not present, its value is interpreted as "None".

The following is the parent element of the **BackFrame.FrameStyle** element.

Parent elements
BackFrame

The following is the XML Schema definition of the **BackFrame.FrameStyle** element.

```
<xsd:element name="FrameStyle" type="xsd:string" minOccurs="0" />
```

### 2.164.5 BackFrame.FrameWidth

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **BackFrame.FrameWidth** element specifies the width (thickness) of a [BackFrame](#). The value of this element is measured as a percentage of the width or height of the containing element. This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be equal to or greater than 0 and equal to or less than 50. If this element is not present, its value is interpreted as 8.

The following is the parent element of the **BackFrame.FrameWidth** element.

Parent elements
BackFrame

The following is the XML Schema definition of the **BackFrame.FrameWidth** element.

```
<xsd:element name="FrameWidth" type="xsd:string" minOccurs="0" />
```

### 2.164.6 BackFrame.GlassEffect

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **BackFrame.GlassEffect** element specifies the glass effect that is applied to a [BackFrame](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**None:** Specifies that there is no glass effect.

**Simple:** Specifies that the glass effect is a simple reflection.

If this element is not present, its value is interpreted as "None".

The following is the parent element of the **BackFrame.GlassEffect** element.

Parent elements
BackFrame

The following is the XML Schema definition of the **BackFrame.GlassEffect** element.

```
<xsd:element name="GlassEffect" type="xsd:string" minOccurs="0" />
```

## 2.164.7 BackFrame.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **BackFrame.Style** element specifies style properties for a [BackFrame](#). The **BackFrame.Style** element is optional. This element is of type [Style](#).

The following is the parent element of the **BackFrame.Style** element.

Parent elements
BackFrame

The following is the XML Schema definition of the **BackFrame.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.165 FrameBackground

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **FrameBackground** element specifies the style of the background of a [LinearGauge](#), a [RadialGauge](#), or a [GaugePanel](#).

The following are the parent and child elements of the **FrameBackground** element.

Parent elements
<a href="#">BackFrame</a>

Child elements
<a href="#">FrameBackground.Style</a>

The following is the XML Schema definition of the **FrameBackground** element.

```
<xsd:complexType name="FrameBackgroundType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

## 2.165.1 FrameBackground.Style

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **FrameBackground.Style** element specifies style properties for a [FrameBackground](#). The **FrameBackground.Style** element is of type [Style](#). This element is optional.

The following is the parent element of the **FrameBackground.Style** element.

Parent elements
<a href="#">FrameBackground</a>

The following is the XML Schema definition of the **FrameBackground.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.166 FrameImage

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **FrameImage** element specifies an image to be used as the background for a [LinearGauge](#), a [RadialGauge](#), or a [GaugePanel](#).

The following are the parent and child elements of the **FrameImage** element.

Parent elements
<a href="#">BackFrame</a>

Child elements
<a href="#">FrameImage.MIMETYPE</a>
<a href="#">FrameImage.Source</a>
<a href="#">FrameImage.TransparentColor</a>
<a href="#">FrameImage.Value</a>
<a href="#">FrameImage.ClipImage</a>
<a href="#">FrameImage.HueColor</a>
<a href="#">FrameImage.Transparency</a>

The following is the XML Schema definition of the **FrameImage** element in RDL 2008/01.

```
<xsd:complexType name="FrameImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

```

    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <xsd:element name="ClipImage" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **FrameImage** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="FrameImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <xsd:element name="ClipImage" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.166.1 FrameImage.MIMETYPE

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **FrameImage.MIMETYPE** element specifies the image format of a [FrameImage](#) instance. The **FrameImage.MIMETYPE** element is optional. If this element is present, its value **MUST** be a [ReportMIMETYPE](#).

If the peer [FrameImage.Source](#) element is set to a value other than "Database", the **FrameImage.MIMETYPE** element is ignored.

The following is the parent element of the **FrameImage.MIMETYPE** element.

Parent elements
FrameImage

The following is the XML Schema definition of the **FrameImage.MIMETYPE** element.

```

<xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />

```

### 2.166.2 FrameImage.Source

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **FrameImage.Source** element specifies the type of source that is associated with a [FrameImage](#) instance. The value of the **FrameImage.Source** element **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element **MUST** be one of the following or an expression that evaluates to one of the following:

**External:** Specifies that the peer [FrameImage.Value](#) element contains a **String** constant or expression that evaluates to the location of an image.

**Embedded:** Specifies that the peer **FrameImage.Value** element contains a **String** constant or expression that evaluates to the name of an EmbeddedImage instance within the report.

**Database:** Specifies that the peer **FrameImage.Value** element contains an expression (for example, a field in the database) that evaluates to the binary data for an image.

The **FrameImage.Source** element MUST be specified.

The following is the parent element of the **FrameImage.Source** element.

Parent elements
FrameImage

The following is the XML Schema definition of the **FrameImage.Source** element.

```
<xsd:element name="Source" type="xsd:string" minOccurs="1" />
```

### 2.166.3 FrameImage.TransparentColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **FrameImage.TransparentColor** element specifies the color to be treated as transparent in a [FrameImage](#) instance. The **FrameImage.TransparentColor** element is optional. The value of this element MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **FrameImage.TransparentColor** element.

Parent elements
FrameImage

The following is the XML Schema definition of the **FrameImage.TransparentColor** element.

```
<xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
```

### 2.166.4 FrameImage.Value

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **FrameImage.Value** element depends on the peer [FrameImage.Source](#) element. The **FrameImage.Value** element MUST be specified.

If the peer **FrameImage.Source** element is set to "External" and the value of the **FrameImage.Value** element is not empty, then the value of the **FrameImage.Value** element MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to the location of an image. This location MUST be a [ReportPath](#) or [RdlURL](#) value.

If the peer **FrameImage.Source** element is set to "Embedded" and the value of the **FrameImage.Value** element is not empty, then the value of the **FrameImage.Value** element MUST be a **String** or an expression that evaluates to the name of an [EmbeddedImage](#) in the report.

If the peer **FrameImage.Source** element is set to "Database" and the value of the **FrameImage.Value** element is not empty, then the value of the **FrameImage.Value** element MUST be an expression that evaluates to the binary data for an image.

If the **FrameImage.Value** element has an empty value, the image MUST NOT be displayed.

The following is the parent element of the **FrameImage.Value** element.

Parent elements
<a href="#">FrameImage</a>

The following is the XML Schema definition of the **FrameImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1" />
```

### 2.166.5 FrameImage.ClipImage

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **FrameImage.ClipImage** element indicates whether a [FrameImage](#) instance is clipped by the physical bounds of the [LinearGauge](#), [RadialGauge](#), or [GaugePanel](#). The **FrameImage.ClipImage** element is optional.

If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **FrameImage.ClipImage** element.

Parent elements
<a href="#">FrameImage</a>

The following is the XML Schema definition of the **FrameImage.ClipImage** element.

```
<xsd:element name="ClipImage" type="xsd:string" minOccurs="0" />
```

### 2.166.6 FrameImage.HueColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **FrameImage.HueColor** element specifies the color with which to tint a [FrameImage](#). The value of the **FrameImage.HueColor** element MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. The **FrameImage.HueColor** element is optional.

The following is the parent element of the **FrameImage.HueColor** element.

Parent elements
<a href="#">FrameImage</a>

The following is the XML Schema definition of the **FrameImage.HueColor** element.

```
<xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
```

## 2.166.7 FrameImage.Transparency

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **FrameImage.Transparency** element specifies the percentage of transparency for a [FrameImage](#). The **FrameImage.Transparency** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **FrameImage.Transparency** element.

Parent elements
<a href="#">FrameImage</a>

The following is the XML Schema definition of the **FrameImage.Transparency** element.

```
<xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
```

## 2.167 GaugeImages

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeImages** element is ignored.

The following are the parent and child elements of the **GaugeImages** element.

Parent elements
<a href="#">GaugePanel</a>

Child elements
<a href="#">GaugeImages.GaugeImage</a>

The following is the XML Schema definition of the **GaugeImages** element in RDL 2008/01.

```
<xsd:complexType name="GaugeImagesType">
  <xsd:sequence>
    <xsd:element name="GaugeImage" type="GaugeImageType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **GaugeImages** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="GaugeImagesType">
  <xsd:sequence>
    <xsd:element name="GaugeImage" type="GaugeImageType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

</xsd:complexType>

### 2.167.1 GaugeImages.GaugeImage

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeImages.GaugeImage** element is ignored. The **GaugeImages.GaugeImage** element is of type [GaugeImage](#).

The following is the parent element of the **GaugeImages.GaugeImage** element.

Parent elements
<a href="#">GaugeImage</a>

The following is the XML Schema definition of the **GaugeImages.GaugeImage** element.

```
<xsd:element name="GaugeImage" type="GaugeImageType" minOccurs="1"
maxOccurs="unbounded" />
```

### 2.168 GaugeImage

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeImage** element is ignored.

The following are the parent element, attribute, and child elements of the **GaugeImage** element.

Parent elements
<a href="#">GaugeImages</a>

Attributes
<a href="#">GaugeImage.Name</a>

Child elements
<a href="#">GaugeImage.ActionInfo</a>
<a href="#">GaugeImage.Height</a>
<a href="#">GaugeImage.Hidden</a>
<a href="#">GaugeImage.Left</a>
<a href="#">GaugeImage.ParentItem</a>
<a href="#">GaugeImage.ToolTip</a>
<a href="#">GaugeImage.Top</a>



Child elements
<a href="#">GaugeImage.Width</a>
<a href="#">GaugeImage.ZIndex</a>
<a href="#">GaugeImage.Angle</a>
<a href="#">GaugeImage.MIMETYPE</a>
<a href="#">GaugeImage.ResizeMode</a>
<a href="#">GaugeImage.Source</a>
<a href="#">GaugeImage.Transparency</a>
<a href="#">GaugeImage.TransparentColor</a>
<a href="#">GaugeImage.Value</a>

The following is the XML Schema definition of the **GaugeImage** element in RDL 2008/01.

```
<xsd:complexType name="GaugeImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **GaugeImage** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="GaugeImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
  </xsd:choice>
</xsd:complexType>
```

```

<xsd:element name="Source" type="xsd:string" minOccurs="1" />
<xsd:element name="Value" type="xsd:string" minOccurs="1" />
<xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
<xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
<xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
<xsd:element name="Angle" type="xsd:string" minOccurs="0" />
<xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.168.1 GaugeImage.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeImage.Name** attribute specifies a unique identifier for a [GaugeImage](#). The **GaugeImage.Name** attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [UTR15].

The following is the parent element of the **GaugeImage.Name** attribute.

Parent elements
<a href="#">GaugeImage</a>

The following is the XML Schema definition of the **GaugeImage.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.168.2 GaugeImage.ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeImage.ActionInfo** element is ignored. This element is of type [ActionInfo](#).

The following is the parent element of the **GaugeImage.ActionInfo** element.

Parent elements
<a href="#">GaugeImage</a>

The following is the XML Schema definition of the **GaugeImage.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.168.3 GaugeImage.Height

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeImage.Height** element is ignored if it is present. However, its data type is validated and the value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**.

The following is the parent element of the **GaugeImage.Height** element.

<b>Parent elements</b>
------------------------

<a href="#">GaugeImage</a>
----------------------------

The following is the XML Schema definition of the **GaugeImage.Height** element.

```
<xsd:element name="Height" type="xsd:string" minOccurs="0" />
```

#### 2.168.4 GaugeImage.Hidden

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeImage.Hidden** element is ignored if it is present. However, its data type is validated and the value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**.

The following is the parent element of the **GaugeImage.Hidden** element.

<b>Parent elements</b>
------------------------

<a href="#">GaugeImage</a>
----------------------------

The following is the XML Schema definition of the **GaugeImage.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

#### 2.168.5 GaugeImage.Left

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeImage.Left** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**.

The following is the parent element of the **GaugeImage.Left** element.

<b>Parent elements</b>
------------------------

<a href="#">GaugeImage</a>
----------------------------

The following is the XML Schema definition of the **GaugeImage.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

#### 2.168.6 GaugeImage.ParentItem

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeImage.ParentItem** element is ignored.

The following is the parent element of the **GaugeImage.ParentItem** element.

<b>Parent elements</b>
------------------------

<a href="#">GaugeImage</a>
----------------------------

The following is the XML Schema definition of the **GaugeImage.ParentItem** element.

```
<xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
```

### 2.168.7 GaugeImage.ToolTip

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeImage.ToolTip** element is ignored. If it is present, however, its data type is validated, and the value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **GaugeImage.ToolTip** element.

<b>Parent elements</b>
------------------------

<a href="#">GaugeImage</a>
----------------------------

The following is the XML Schema definition of the **GaugeImage.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

### 2.168.8 GaugeImage.Top

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeImage.Top** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**.

The following is the parent element of the **GaugeImage.Top** element.

<b>Parent elements</b>
------------------------

<a href="#">GaugeImage</a>
----------------------------

The following is the XML Schema definition of the **GaugeImage.Top** element.

```
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
```

### 2.168.9 GaugeImage.Width

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeImage.Width** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**.

The following is the parent element of the **GaugeImage.Width** element.

<b>Parent elements</b>
------------------------

<a href="#">GaugeImage</a>
----------------------------

The following is the XML Schema definition of the **GaugeImage.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
```

### 2.168.10 GaugeImage.ZIndex

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeImage.ZIndex** element is ignored if it is present. However, its data type is validated, and the value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**.

The following is the parent element of the **GaugeImage.ZIndex** element.

<b>Parent elements</b>
------------------------

<a href="#">GaugeImage</a>
----------------------------

The following is the XML Schema definition of the **GaugeImage.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
```

### 2.168.11 GaugeImage.Angle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeImage.Angle** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**.

The following is the parent element of the **GaugeImage.Angle** element.

<b>Parent elements</b>
------------------------

<a href="#">GaugeImage</a>
----------------------------

The following is the XML Schema definition of the **GaugeImage.Angle** element.

```
<xsd:element name="Angle" type="xsd:string" minOccurs="0" />
```

### 2.168.12 GaugeImage.MIMETYPE

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeImage.MIMETYPE** element is ignored if it is present. However, its data type is validated, and the value MUST be a [ReportMIMETYPE](#) or an expression that evaluates to a **ReportMIMETYPE**.

The following is the parent element of the **GaugeImage.MIMETYPE** element.

<b>Parent elements</b>
------------------------

<a href="#">GaugeImage</a>
----------------------------

The following is the XML Schema definition of the **GaugeImage.MIMETYPE** element.

```
<xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
```

### 2.168.13 GaugeImage.ResizeMode

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeImage.ResizeMode** element is ignored if it is present. However, its data type is validated and this value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

- **AutoFit**
- **None**

The following is the parent element of the **GaugeImage.ResizeMode** element.

<b>Parent elements</b>
------------------------

<a href="#">GaugeImage</a>
----------------------------

The following is the XML Schema definition of the **GaugeImage.ResizeMode** element.

```
<xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
```

### 2.168.14 GaugeImage.Source

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeImage.Source** element is ignored if it is present. However, its data type is validated and this value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

- **External**
- **Embedded**
- **Database**

The **GaugeImage.Source** element MUST be specified.

The following is the parent element of the **GaugeImage.Source** element.

<b>Parent elements</b>
------------------------

<a href="#">GaugeImage</a>
----------------------------

The following is the XML Schema definition of the **GaugeImage.Source** element.

```
<xsd:element name="Source" type="xsd:string" minOccurs="1" />
```

### 2.168.15 GaugeImage.Transparency

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeImage.Transparency** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**.

The following is the parent element of the **GaugeImage.Transparency** element.

Parent elements
<a href="#">GaugeImage</a>

The following is the XML Schema definition of the **GaugeImage.Transparency** element.

```
<xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
```

### 2.168.16 GaugeImage.TransparentColor

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeImage.TransparentColor** element is ignored if it is present. However, its data type is validated, and the value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **GaugeImage.TransparentColor** element.

Parent elements
<a href="#">GaugeImage</a>

The following is the XML Schema definition of the **GaugeImage.TransparentColor** element.

```
<xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
```

### 2.168.17 GaugeImage.Value

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeImage.Value** element is ignored.

The following is the parent element of the **GaugeImage.Value** element.

Parent elements
<a href="#">GaugeImage</a>

The following is the XML Schema definition of the **GaugeImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1" />
```

## 2.169 GaugeLabels

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeLabels** element specifies the set of [GaugeLabel](#) instances for a [GaugePanel](#). The **GaugeLabels** element is optional. If this element is present, it MUST contain at least one [GaugeLabels.GaugeLabel](#) instance.

The following are the parent and child elements of the **GaugeLabels** element.

Parent elements
GaugePanel

Child elements
GaugeLabels.GaugeLabel

The following is the XML Schema definition of the **GaugeLabels** element in RDL 2008/01.

```
<xsd:complexType name="GaugeLabelsType">
  <xsd:sequence>
    <xsd:element name="GaugeLabel" type="GaugeLabelType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **GaugeLabels** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="GaugeLabelsType">
  <xsd:sequence>
    <xsd:element name="GaugeLabel" type="GaugeLabelType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.169.1 GaugeLabels.GaugeLabel

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeLabels.GaugeLabel** element specifies a [GaugeLabel](#) within the set of [GaugeLabels](#) for a [GaugePanel](#). The **GaugeLabels.GaugeLabel** element MUST be specified. This element is of type **GaugeLabel**.

The following is the parent element of the **GaugeLabels.GaugeLabel** element.

Parent elements
GaugeLabels

The following is the XML Schema definition of the **GaugeLabels.GaugeLabel** element.



```
<xsd:element name="GaugeLabel" type="GaugeLabelType" minOccurs="1"
maxOccurs="unbounded" />
```

## 2.170 GaugeLabel

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeLabel** element specifies a label to display within a [GaugePanel](#) instance. The **GaugeLabel** element **MUST** be specified at least once within a [GaugeLabels](#) collection.

The following are the parent elements, attributes, and child elements of the **GaugeLabel** element.

Parent elements
<a href="#">GaugeLabels</a>

Attributes
<a href="#">GaugeLabel.Name</a>

Child elements
<a href="#">GaugeLabel.ActionInfo</a>
<a href="#">GaugeLabel.Height</a>
<a href="#">GaugeLabel.Hidden</a>
<a href="#">GaugeLabel.Left</a>
<a href="#">GaugeLabel.ParentItem</a>
<a href="#">GaugeLabel.ToolTip</a>
<a href="#">GaugeLabel.Top</a>
<a href="#">GaugeLabel.Width</a>
<a href="#">GaugeLabel.ZIndex</a>
<a href="#">GaugeLabel.Angle</a>
<a href="#">GaugeLabel.ResizeMode</a>
<a href="#">GaugeLabel.Style</a>
<a href="#">GaugeLabel.Text</a>
<a href="#">GaugeLabel.TextShadowOffset</a>
<a href="#">GaugeLabel.UseFontPercent</a>

The following is the XML Schema definition of the **GaugeLabel** element in RDL 2008/01.

```
<xsd:complexType name="GaugeLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
```

```

<!--GaugePanelItemTypeStart-->
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
<xsd:element name="Height" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
<!--GaugePanelItemTypeEnd-->
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="Text" type="xsd:string" minOccurs="0" />
<xsd:element name="Angle" type="xsd:string" minOccurs="0" />
<xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
<xsd:element name="TextShadowOffset" type="xsd:string" minOccurs="0" />
<xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **GaugeLabel** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="GaugeLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextShadowOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

## 2.170.1 GaugeLabel.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeLabel.Name** attribute specifies a unique identifier for a [GaugeLabel](#). The **GaugeLabel.Name** attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [UTR15] that is unique for a particular [GaugeLabels](#) collection.

The following is the parent element of the **GaugeLabel.Name** attribute.

Parent elements
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

## 2.170.2 GaugeLabel.ActionInfo

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.ActionInfo** element specifies the actions for a [GaugeLabel](#). The **GaugeLabel.ActionInfo** element is optional. This element is of type [ActionInfo](#).

The following is the parent element of the **GaugeLabel.ActionInfo** element.

Parent elements
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

## 2.170.3 GaugeLabel.Height

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.Height** element specifies the height of a [GaugeLabel](#) as a percentage of [GaugeLabel.ParentItem](#). If the peer element **GaugeLabel.ParentItem** is not specified, the value of the **GaugeLabel.Height** element is interpreted as relative to the height of the [GaugePanel](#).

The **GaugeLabel.Height** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If the [GaugePanel.AutoLayout](#) property for the **GaugeLabel.Height** element's parent **GaugePanel** element is set to true, the **GaugeLabel.Height** element is ignored. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **GaugeLabel.Height** element.

Parent elements
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.Height** element.

```
<xsd:element name="Height" type="xsd:string" minOccurs="0" />
```

## 2.170.4 GaugeLabel.Hidden

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.Hidden** element specifies whether a [GaugeLabel](#) is hidden. The **GaugeLabel.Hidden** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **GaugeLabel.Hidden** element.

Parent elements
<a href="#">GaugeLabel</a>

The following is the XML Schema definition of the **GaugeLabel.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

### 2.170.5 GaugeLabel.Left

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.Left** element specifies the distance from the left as a percentage of [GaugeLabel.ParentItem](#). If **GaugeLabel.ParentItem** is not specified, the value of the **GaugeLabel.Left** element is interpreted as relative to the left of the [GaugePanel](#).

The **GaugeLabel.Left** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If the [GaugePanel.AutoLayout](#) property for the **GaugeLabel.Left** element's parent **GaugePanel** element is set to true, the **GaugeLabel.Left** element is ignored. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **GaugeLabel.Left** element.

Parent elements
<a href="#">GaugeLabel</a>

The following is the XML Schema definition of the **GaugeLabel.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

### 2.170.6 GaugeLabel.ParentItem

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.ParentItem** element specifies the name of the parent [GaugeLabel](#). The **GaugeLabel.ParentItem** element is optional.

The following is the parent element of the **GaugeLabel.ParentItem** element.

Parent elements
<a href="#">GaugeLabel</a>

The following is the XML Schema definition of the **GaugeLabel.ParentItem** element.

```
<xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
```

### 2.170.7 GaugeLabel.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.ToolTip** element specifies the tooltip text for a [GaugeLabel](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **GaugeLabel.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **GaugeLabel.ToolTip** element.

Parent elements
<a href="#">GaugeLabel</a>

The following is the XML Schema definition of the **GaugeLabel.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

### 2.170.8 GaugeLabel.Top

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.Top** element specifies the distance from the top as a percentage of [GaugeLabel.ParentItem](#). If **GaugeLabel.ParentItem** is not specified, the value of **GaugeLabel.Top** is interpreted as relative to the top of the [GaugePanel](#).

The **GaugeLabel.Top** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If the [GaugePanel.AutoLayout](#) property for the **GaugeLabel.Top** element's parent **GaugePanel** element is set to true, the **GaugeLabel.Top** element is ignored. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **GaugeLabel.Top** element.

Parent elements
<a href="#">GaugeLabel</a>

The following is the XML Schema definition of the **GaugeLabel.Top** element.

```
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
```

### 2.170.9 GaugeLabel.Width

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.Width** element specifies the width of the [GaugeLabel](#) as a percentage of [GaugeLabel.ParentItem](#). If **GaugeLabel.ParentItem** is not specified, the value of **GaugeLabel.Width** is interpreted as relative to the width of the [GaugePanel](#).

The **GaugeLabel.Width** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If the [GaugePanel.AutoLayout](#) property for the **GaugeLabel.Width** element's parent **GaugePanel** element is set to true, the **GaugeLabel.Width** element is ignored. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **GaugeLabel.Width** element.

Parent elements
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
```

### 2.170.10 GaugeLabel.ZIndex

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.ZIndex** element specifies the drawing order of a [GaugeLabel](#) within a [GaugePanel](#). The **GaugeLabel.ZIndex** element is optional. If this element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 0. The value of this element MUST be greater than or equal to 0 and less than or equal to 2147483647.

The following is the parent element of the **GaugeLabel.ZIndex** element.

Parent elements
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
```

### 2.170.11 GaugeLabel.Angle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.Angle** element specifies the degrees for the angle of rotation for a [GaugeLabel](#). The **GaugeLabel.Angle** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. The **Float** value MUST NOT be less than 0 or greater than 360. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **GaugeLabel.Angle** element.

Parent elements
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.Angle** element.

```
<xsd:element name="Angle" type="xsd:string" minOccurs="0">
```

## 2.170.12 GaugeLabel.ResizeMode

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.ResizeMode** element specifies whether content will resize to fit in the available space of a [GaugeLabel](#). The **GaugeLabel.ResizeMode** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The specified **String** value for this element MUST be one of the following:

**AutoFit**: The content will automatically resize to fit in the available space of the label.

**None**: The content will not automatically resize to fit in the available space of the label.

If the **GaugeLabel.ResizeMode** element is not present, its value is interpreted as "AutoFit".

The following is the parent element of the **GaugeLabel.ResizeMode** element.

Parent elements
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.ResizeMode** element.

```
<xsd:element name="ResizeMode" type="xsd:string" minOccurs="0">
```

## 2.170.13 GaugeLabel.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.Style** element specifies style properties for a [GaugeLabel](#). The **GaugeLabel.Style** element is optional. This element is of type [Style](#).

The following is the parent element of the **GaugeLabel.Style** element.

Parent elements
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0">
```

## 2.170.14 GaugeLabel.Text

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.Text** element specifies the text content of a [GaugeLabel](#). The **GaugeLabel.Text** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The following is the parent element of the **GaugeLabel.Text** element.

Parent elements
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.Text** element.

```
<xsd:element name="Text" type="xsd:string" minOccurs="0">
```

### 2.170.15 GaugeLabel.TextShadowOffset

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.TextShadowOffset** element specifies the diagonal bottom-right positional offset of the text shadow from the [GaugeLabel.Text](#) within a [GaugeLabel](#). The **GaugeLabel.TextShadowOffset** element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**.

The following is the parent element of the **GaugeLabel.TextShadowOffset** element.

Parent elements
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.TextShadowOffset** element.

```
<xsd:element name="TextShadowOffset" type="xsd:string" minOccurs="0">
```

### 2.170.16 GaugeLabel.UseFontPercent

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeLabel.UseFontPercent** element specifies whether the font size in a [GaugeLabel](#) is measured as a percentage of the parent element or in units specified by the [Style.FontSize](#) element.

The **GaugeLabel.UseFontPercent** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **GaugeLabel.UseFontPercent** element.

Parent elements
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.UseFontPercent** element.

```
<xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0">
```

### 2.171 GaugeMember

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeMember** element specifies group, sort, and filter behavior for the data within a [GaugePanel](#) or another **GaugeMember**. The **GaugeMember** element MUST contain at least one [GaugeMember.Group](#) instance. This element is optional.

The following are the parent and child elements of the **GaugeMember** element.



<b>Parent elements</b>
GaugePanel
<b>GaugeMember</b>

<b>Child elements</b>
<a href="#">GaugeMember.GaugeMember</a>
GaugeMember.Group
<a href="#">GaugeMember.SortExpressions</a>

The following is the XML Schema definition of the **GaugeMember** element in RDL 2008/01.

```
<xsd:complexType name="GaugeMemberType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="1" />
    <xsd:element name="SortExpressions" type="SortExpressionsType"
      minOccurs="0" />
    <xsd:element name="GaugeMember" type="GaugeMemberType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **GaugeMember** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="GaugeMemberType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="1" />
    <xsd:element name="SortExpressions" type="SortExpressionsType"
      minOccurs="0" />
    <xsd:element name="GaugeMember" type="GaugeMemberType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.171.1 GaugeMember.GaugeMember

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeMember.GaugeMember** element specifies nested grouping/sorting filtering behavior for data within a [GaugePanel](#). The **GaugeMember.GaugeMember** element is optional. This element is of type [GaugeMember](#).

The following is the parent element of the **GaugeMember.GaugeMember** element.

<b>Parent elements</b>
GaugeMember

The following is the XML Schema definition of the **GaugeMember.GaugeMember** element.

```
<xsd:element name="GaugeMember" type="GaugeMemberType" minOccurs="0">
```

## 2.171.2 GaugeMember.Group

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeMember.Group** element specifies grouping to apply to the data within a [GaugePanel](#). The **GaugeMember.Group** element MUST be specified. This element is of type [Group](#).

The following is the parent element of the **GaugeMember.Group** element.

Parent elements
<a href="#">GaugeMember</a>

The following is the XML Schema definition of the **GaugeMember.Group** element.

```
<xsd:element name="Group" type="GroupType" minOccurs="1">
```

## 2.171.3 GaugeMember.SortExpressions

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeMember.SortExpressions** element specifies sorting to apply to the groups that are defined within the peer element [GaugeMember.Group](#). The **GaugeMember.SortExpressions** element is optional. This element is of type [SortExpressions](#).

The following is the parent element of the **GaugeMember.SortExpressions** element.

Parent elements
<a href="#">GaugeMember</a>

The following is the XML Schema definition of the **GaugeMember.SortExpressions** element.

```
<xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
```

## 2.172 LinearGauges

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearGauges** element specifies the set of [LinearGauge](#) instances for a [GaugePanel](#). The **LinearGauges** element is optional. If this element is present, it MUST contain at least one [LinearGauges.LinearGauge](#) instance.

The following are the parent and child elements of the **LinearGauges** element.

Parent elements
GaugePanel

Child elements
----------------

LinearGauges.LinearGauge
--------------------------

The following is the XML Schema definition of the **LinearGauges** element in RDL 2008/01.

```
<xsd:complexType name="LinearGaugesType">
  <xsd:sequence>
    <xsd:element name="LinearGauge" type="LinearGaugeType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **LinearGauges** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="LinearGaugesType">
  <xsd:sequence>
    <xsd:element name="LinearGauge" type="LinearGaugeType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.172.1 LinearGauges.LinearGauge

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearGauges.LinearGauge** element specifies a [LinearGauge](#) within the collection of [LinearGauges](#) for a [GaugePanel](#). This element MUST be specified at least once within a **LinearGauges** collection. The **LinearGauges.LinearGauge** element is of type **LinearGauge**.

The following is the parent element of the **LinearGauges.LinearGauge** element.

Parent elements
-----------------

LinearGauges
--------------

The following is the XML Schema definition of the **LinearGauges.LinearGauge** element.

```
<xsd:element name="LinearGauge" type="LinearGaugeType" minOccurs="1"
  maxOccurs="unbounded" />
```

### 2.173 LinearGauge

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearGauge** element specifies a **LinearGauge** to be drawn within a [GaugePanel](#). This element MUST be specified at least once within a [LinearGauges](#) collection.

The following are the parent elements, attributes, and child elements of the **LinearGauge** element.

Parent elements
-----------------

LinearGauges
--------------

Attributes
<a href="#">LinearGauge.Name</a>

Child elements
<a href="#">LinearGauge.ActionInfo</a>
<a href="#">LinearGauge.Height</a>
<a href="#">LinearGauge.Hidden</a>
<a href="#">LinearGauge.Left</a>
<a href="#">LinearGauge.ParentItem</a>
<a href="#">LinearGauge.ToolTip</a>
<a href="#">LinearGauge.Top</a>
<a href="#">LinearGauge.Width</a>
<a href="#">LinearGauge.ZIndex</a>
<a href="#">LinearGauge.AspectRatio</a>
<a href="#">LinearGauge.BackFrame</a>
<a href="#">LinearGauge.ClipContent</a>
<a href="#">LinearGauge.GaugeScales</a>
<a href="#">LinearGauge.TopImage</a>
<a href="#">LinearGauge.Orientation</a>

The following is the XML Schema definition of the **LinearGauge** element in RDL 2008/01.

```

<xsd:complexType name="LinearGaugeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeTypeStart-->
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
    <xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="ClipContent" type="xsd:string" minOccurs="0" />
    <xsd:element name="AspectRatio" type="xsd:string" minOccurs="0" />
    <!--GaugeTypeEnd-->
    <xsd:element name="GaugeScales" type="LinearScalesType"
      minOccurs="0" />
    <xsd:element name="Orientation" type="xsd:string" minOccurs="0" />
  
```

```

        <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **LinearGauge** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="LinearGaugeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeTypeStart-->
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
    <xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="ClipContent" type="xsd:string" minOccurs="0" />
    <xsd:element name="AspectRatio" type="xsd:string" minOccurs="0" />
    <!--GaugeTypeEnd-->
    <xsd:element name="GaugeScales" type="LinearScalesType"
      minOccurs="0" />
    <xsd:element name="Orientation" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.173.1 LinearGauge.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearGauge.Name** attribute specifies a unique identifier for a [LinearGauge](#). This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique within a particular [LinearGauges](#) collection.

The following is the parent element of the **LinearGauge.Name** attribute.

Parent elements
LinearGauge

The following is the XML Schema definition of the **LinearGauge.Name** attribute.

```

<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />

```

### 2.173.2 LinearGauge.ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearGauge.ActionInfo** element specifies the actions for a [LinearGauge](#). This element is optional. This element is of type [ActionInfo](#).

The following is the parent element of the **LinearGauge.ActionInfo** element.

Parent elements
LinearGauge

The following is the XML Schema definition of the **LinearGauge.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.173.3 LinearGauge.Height

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearGauge.Height** element specifies the height of a [LinearGauge](#) as a percentage of [LinearGauge.ParentItem](#). If the **LinearGauge.ParentItem** element is not specified, the value of the **LinearGauge.Height** element is interpreted as relative to the height of the [GaugePanel](#).

The **LinearGauge.Height** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. If the [GaugePanel.AutoLayout](#) property for the **LinearGauge.Height** element's parent **GaugePanel** element is set to true, the **LinearGauge.Height** element is ignored. If the **LinearGauge.Height** element is not present, its value is interpreted as 0.

The following is the parent element of the **LinearGauge.Height** element.

Parent elements
LinearGauge

The following is the XML Schema definition of the **LinearGauge.Height** element.

```
<xsd:element name="Height" type="xsd:string" minOccurs="0" />
```

### 2.173.4 LinearGauge.Hidden

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearGauge.Hidden** element specifies whether a [LinearGauge](#) is hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **LinearGauge.Hidden** element.

Parent elements
LinearGauge

The following is the XML Schema definition of the **LinearGauge.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

### 2.173.5 LinearGauge.Left

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearGauge.Left** element specifies the distance from the left as a percentage of [LinearGauge.ParentItem](#). If **LinearGauge.ParentItem** is not present, the value of the **LinearGauge.Left** element is interpreted as relative to the left of the [GaugePanel](#).

The **LinearGauge.Left** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If the [GaugePanel.AutoLayout](#) property for the **LinearGauge.Left** element's parent **GaugePanel** element is set to true, the **LinearGauge.Left** element is ignored. If the **LinearGauge.Left** element is not present, its value is interpreted as 0.

The following is the parent element of the **LinearGauge.Left** element.

Parent elements
<a href="#">LinearGauge</a>

The following is the XML Schema definition of the **LinearGauge.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

### 2.173.6 LinearGauge.ParentItem

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearGauge.ParentItem** element specifies the name of the parent [LinearGauge](#). The **LinearGauge.ParentItem** element is optional.

The following is the parent element of the **LinearGauge.ParentItem** element.

Parent elements
<a href="#">LinearGauge</a>

The following is the XML Schema definition of the **LinearGauge.ParentItem** element.

```
<xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
```

### 2.173.7 LinearGauge.ToolTip

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearGauge.ToolTip** element specifies the tooltip text for a [LinearGauge](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **LinearGauge.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **LinearGauge.ToolTip** element.

Parent elements
LinearGauge

The following is the XML Schema definition of the **LinearGauge.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

### 2.173.8 LinearGauge.Top

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearGauge.Top** element specifies the distance from the top as a percentage of [LinearGauge.ParentItem](#). If **LinearGauge.ParentItem** is not specified, the value of the **LinearGauge.Top** element is interpreted as relative to the top of the [GaugePanel](#).

The **LinearGauge.Top** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If the [GaugePanel.AutoLayout](#) property for the **LinearGauge.Top** element's parent **GaugePanel** element is set to true, the **LinearGauge.Top** element is ignored. If the **LinearGauge.Top** element is not present, its value is interpreted as 0.

The following is the parent element of the **LinearGauge.Top** element.

Parent elements
<a href="#">LinearGauge</a>

The following is the XML Schema definition of the **LinearGauge.Top** element.

```
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
```

### 2.173.9 LinearGauge.Width

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearGauge.Width** element specifies the width of a [LinearGauge](#) as a percentage of [LinearGauge.ParentItem](#). If **LinearGauge.ParentItem** is not specified, the value of the **LinearGauge.Width** element is interpreted as relative to the width of the [GaugePanel](#).

The **LinearGauge.Width** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If the [GaugePanel.AutoLayout](#) property for the **LinearGauge.Width** element's parent **GaugePanel** element is set to true, the **LinearGauge.Width** element is ignored. If the **LinearGauge.Width** element is not present, its value is interpreted as 0.

The following is the parent element of the **LinearGauge.Width** element.

Parent elements
LinearGauge

The following is the XML Schema definition of the **LinearGauge.Width** element.



```
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
```

### 2.173.10 LinearGauge.ZIndex

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearGauge.ZIndex** element specifies the drawing order of a [LinearGauge](#) within a [GaugePanel](#). This element is optional. If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 0. The value of the **LinearGauge.ZIndex** element MUST be greater than or equal to 0 and less than or equal to 2147483647.

The following is the parent element of the **LinearGauge.ZIndex** element.

Parent elements
LinearGauge

The following is the XML Schema definition of the **LinearGauge.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
```

### 2.173.11 LinearGauge.AspectRatio

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearGauge.AspectRatio** element specifies the ratio of the height to the width for drawing a [LinearGauge](#). This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of the **LinearGauge.AspectRatio** element MUST be greater than or equal to 0. If the **Float** value of this element is 0 or if this element is not present, its value is interpreted based on the contents of the **gauge**.

The following is the parent element of the **LinearGauge.AspectRatio** element.

Parent elements
LinearGauge

The following is the XML Schema definition of the **LinearGauge.AspectRatio** element.

```
<xsd:element name="AspectRatio" type="xsd:string" minOccurs="0" />
```

### 2.173.12 LinearGauge.BackFrame

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearGauge.BackFrame** element specifies the frame and frame background properties for a [LinearGauge](#). This element is optional. This element is of type [BackFrame](#).

The following is the parent element of the **LinearGauge.BackFrame** element.

Parent elements
LinearGauge

The following is the XML Schema definition of the **LinearGauge.BackFrame** element.

```
<xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
```

### 2.173.13 LinearGauge.ClipContent

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearGauge.ClipContent** element indicates whether the content of a [LinearGauge](#) is clipped by the bounds or frame of the **linear gauge**. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false. The following is the parent element of the **LinearGauge.ClipContent** element.

Parent elements
LinearGauge

The following is the XML Schema definition of the **LinearGauge.ClipContent** element.

```
<xsd:element name="ClipContent" type="xsd:string" minOccurs="0" />
```

### 2.173.14 LinearGauge.GaugeScales

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearGauge.GaugeScales** element specifies the set of scales to display on a [LinearGauge](#). This element is optional. This element is of type [LinearScales](#).

The following is the parent element of the **LinearGauge.GaugeScales** element.

Parent elements
LinearGauge

The following is the XML Schema definition of the **LinearGauge.GaugeScales** element.

```
<xsd:element name="GaugeScales" type="LinearScalesType" minOccurs="0">
```

### 2.173.15 LinearGauge.TopImage

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearGauge.TopImage** element specifies the image to display over the top of a [LinearGauge](#). This element is optional. This element is of type [TopImage](#).

The following is the parent element of the **LinearGauge.TopImage** element.

Parent elements
LinearGauge

The following is the XML Schema definition of the **LinearGauge.TopImage** element.

```
<xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
```

### 2.173.16 LinearGauge.Orientation

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearGauge.Orientation** element specifies the orientation of a [LinearGauge](#). This element is optional.

If the **LinearGauge.Orientation** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Auto**: Specifies that the orientation is automatically determined based on the width and height of the **linear gauge**.

**Horizontal**: Specifies that the linear gauge is horizontal.

**Vertical**: Specifies that the linear gauge is vertical.

If the **LinearGauge.Orientation** element is not present, its value is interpreted as "Auto".

The following is the parent element of the **LinearGauge.Orientation** element.

Parent elements
LinearGauge

The following is the XML Schema definition of the **LinearGauge.Orientation** element.

```
<xsd:element name="Orientation" type="xsd:string" minOccurs="0" />
```

### 2.174 LinearScales

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScales** element specifies the collection of [LinearScale](#) instances for a [LinearGauge](#). The **LinearScales** element MUST contain at least one [LinearScales.LinearScale](#) instance.

The following are the parent and child elements of the **LinearScales** element.

Parent elements
LinearGauge

<b>Child elements</b>
-----------------------

LinearScales.LinearScale
--------------------------

The following is the XML Schema definition of the **LinearScales** element in RDL 2008/01.

```
<xsd:complexType name="LinearScalesType">
  <xsd:sequence>
    <xsd:element name="LinearScale" type="LinearScaleType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **LinearScales** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="LinearScalesType">
  <xsd:sequence>
    <xsd:element name="LinearScale" type="LinearScaleType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.174.1 LinearScales.LinearScale

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearScales.LinearScale** element specifies a [LinearScale](#) within a collection of [LinearGauge.GaugeScales](#). The **LinearScales.LinearScale** element MUST be specified at least once within a [LinearScales](#) collection. The **LinearScales.LinearScale** element is of type **LinearScale**.

The following is the parent element of the **LinearScales.LinearScale** element.

<b>Parent elements</b>
------------------------

LinearScales
--------------

The following is the XML Schema definition of the **LinearScales.LinearScale** element.

```
<xsd:element name="LinearScale" type="LinearScaleType" minOccurs="1"
  maxOccurs="unbounded" />
```

## 2.175 LinearScale

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearScale** element specifies a **LinearScale** to be drawn within a [LinearGauge](#). This element MUST be specified at least once within a [LinearScales](#) collection.

The following are the parent elements, attributes, and child elements of the **LinearScale** element.

<b>Parent elements</b>
------------------------

LinearScales
--------------

Attributes
<a href="#">LinearScale.Name</a>

Child elements
<a href="#">LinearScale.ActionInfo</a>
<a href="#">LinearScale.CustomLabels</a>
<a href="#">LinearScale.GaugeMajorTickMarks</a>
<a href="#">LinearScale.GaugeMinorTickMarks</a>
<a href="#">LinearScale.GaugePointers</a>
<a href="#">LinearScale.Hidden</a>
<a href="#">LinearScale.Interval</a>
<a href="#">LinearScale.IntervalOffset</a>
<a href="#">LinearScale.Logarithmic</a>
<a href="#">LinearScale.LogarithmicBase</a>
<a href="#">LinearScale.MaximumPin</a>
<a href="#">LinearScale.MaximumValue</a>
<a href="#">LinearScale.MinimumPin</a>
<a href="#">LinearScale.MinimumValue</a>
<a href="#">LinearScale.Multiplier</a>
<a href="#">LinearScale.Reversed</a>
<a href="#">LinearScale.ScaleLabels</a>
<a href="#">LinearScale.ScaleRanges</a>
<a href="#">LinearScale.Style</a>
<a href="#">LinearScale.TickMarksOnTop</a>
<a href="#">LinearScale.ToolTip</a>
<a href="#">LinearScale.Width</a>
<a href="#">LinearScale.EndMargin</a>
<a href="#">LinearScale.Position</a>
<a href="#">LinearScale.StartMargin</a>

The following is the XML Schema definition of the **LinearScale** element in RDL 2008/01.

```
<xsd:complexType name="LinearScaleType">
```

```

<xsd:choice minOccurs="0" maxOccurs="unbounded">
  <!--GaugeScaleTypeStart-->
  <xsd:element name="Style" type="StyleType" minOccurs="0" />
  <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
  <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
  <xsd:element name="ScaleRanges" type="ScaleRangesType" minOccurs="0" />
  <xsd:element name="ScaleLabels" type="ScaleLabelsType" minOccurs="0" />
  <xsd:element name="GaugeMajorTickMarks" type="GaugeTickMarksType"
    minOccurs="0" />
  <xsd:element name="GaugeMinorTickMarks" type="GaugeTickMarksType"
    minOccurs="0" />
  <xsd:element name="CustomLabels" type="CustomLabelsType" minOccurs="0" />
  <xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
  <xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
  <xsd:element name="MaximumPin" type="ScalePinType" minOccurs="0" />
  <xsd:element name="MinimumPin" type="ScalePinType" minOccurs="0" />
  <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
  <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
  <xsd:element name="Logarithmic" type="xsd:string" minOccurs="0" />
  <xsd:element name="LogarithmicBase" type="xsd:string" minOccurs="0" />
  <xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
  <xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
  <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
  <xsd:element name="Width" type="xsd:string" minOccurs="0" />
  <xsd:element name="TickMarksOnTop" type="xsd:string" minOccurs="0" />
  <!--GaugeScaleTypeEnd-->
  <xsd:element name="GaugePointers" type="LinearPointersType"
    minOccurs="0" />
  <xsd:element name="StartMargin" type="xsd:string" minOccurs="0" />
  <xsd:element name="EndMargin" type="xsd:string" minOccurs="0" />
  <xsd:element name="Position" type="xsd:string" minOccurs="0" />
  <xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **LinearScale** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="LinearScaleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeScaleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="ScaleRanges" type="ScaleRangesType" minOccurs="0" />
    <xsd:element name="ScaleLabels" type="ScaleLabelsType" minOccurs="0" />
    <xsd:element name="GaugeMajorTickMarks" type="GaugeTickMarksType"
      minOccurs="0" />
    <xsd:element name="GaugeMinorTickMarks" type="GaugeTickMarksType"
      minOccurs="0" />
    <xsd:element name="CustomLabels" type="CustomLabelsType" minOccurs="0" />
    <xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MaximumPin" type="ScalePinType" minOccurs="0" />
    <xsd:element name="MinimumPin" type="ScalePinType" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="Logarithmic" type="xsd:string" minOccurs="0" />
    <xsd:element name="LogarithmicBase" type="xsd:string" minOccurs="0" />
    <xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="TickMarksOnTop" type="xsd:string" minOccurs="0" />
    <!--GaugeScaleTypeEnd-->

```

```

<xsd:element name="GaugePointers" type="LinearPointersType"
  minOccurs="0" />
<xsd:element name="StartMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="EndMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.175.1 LinearScale.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearScale.Name** attribute specifies a unique identifier for a [LinearScale](#). This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique within a particular [LinearScales](#) collection.

The following is the parent element of the **LinearScale.Name** attribute.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.175.2 LinearScale.ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearScale.ActionInfo** element specifies the actions for a [LinearScale](#). This element is optional. This element is of type [ActionInfo](#).

The following is the parent element of the **LinearScale.ActionInfo** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.175.3 LinearScale.CustomLabels

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearScale.CustomLabels** element specifies the custom labels for a [LinearScale](#). This element is optional. This element is of type [CustomLabels](#).

The following is the parent element of the **LinearScale.CustomLabels** element.

<b>Parent elements</b>
LinearScale

The following is the XML Schema definition of the **LinearScale.CustomLabels** element.

```
<xsd:element name="CustomLabels" type="CustomLabelsType" minOccurs="0" />
```

#### 2.175.4 LinearScale.GaugeMajorTickMarks

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScale.GaugeMajorTickMarks** element specifies the major tick marks to display on a [LinearScale](#). This element is optional. This element is of type [GaugeTickMarks](#).

The following is the parent element of the **LinearScale.GaugeMajorTickMarks** element.

<b>Parent elements</b>
LinearScale

The following is the XML Schema definition of the **LinearScale.GaugeMajorTickMarks** element.

```
<xsd:element name="GaugeMajorTickMarks" type="GaugeTickMarksType" minOccurs="0" />
```

#### 2.175.5 LinearScale.GaugeMinorTickMarks

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScale.GaugeMinorTickMarks** element specifies the minor tick marks to display on a [LinearScale](#). This element is optional. This element is of type [GaugeTickMarks](#).

The following is the parent element of the **LinearScale.GaugeMinorTickMarks** element.

<b>Parent elements</b>
LinearScale

The following is the XML Schema definition of the **LinearScale.GaugeMinorTickMarks** element.

```
<xsd:element name="GaugeMinorTickMarks"
  type="GaugeTickMarksType" minOccurs="0" />
```

#### 2.175.6 LinearScale.GaugePointers

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScale.GaugePointers** element specifies the **linear pointers** ([LinearPointer](#) instances) to display on a [LinearScale](#). This element is optional. This element is of type [LinearPointers](#).

The following is the parent element of the **LinearScale.GaugePointers** element.



Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.GaugePointers** element.

```
<xsd:element name="GaugePointers" type="LinearPointersType" minOccurs="0" />
```

### 2.175.7 LinearScale.Hidden

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearScale.Hidden** element specifies whether a [LinearScale](#) is hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **LinearScale.Hidden** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

### 2.175.8 LinearScale.Interval

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearScale.Interval** element specifies the default interval between [GaugeTickMarks](#) instances and [LinearScale.ScaleLabels](#) instances within a [LinearScale](#). The **LinearScale.Interval** element is optional.

If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0 and less than or equal to  $1.79769 \times 10^{308}$ . If this element is not present, its value is interpreted as 0.

The following is the parent element of the **LinearScale.Interval** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.Interval** element.

```
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
```

### 2.175.9 LinearScale.IntervalOffset

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearScale.IntervalOffset** element specifies the default offset for the first [GaugeTickMarks](#) instance and [LinearScale.ScaleLabels](#) instance within a [LinearScale](#).

The **LinearScale.IntervalOffset** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be nonnegative. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **LinearScale.IntervalOffset** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.IntervalOffset** element.

```
<xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
```

## 2.175.10 LinearScale.Logarithmic

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScale.Logarithmic** element specifies whether a [LinearScale](#) is logarithmic. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **LinearScale.Logarithmic** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.Logarithmic** element.

```
<xsd:element name="Logarithmic" type="xsd:string" minOccurs="0" />
```

## 2.175.11 LinearScale.LogarithmicBase

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScale.LogarithmicBase** element specifies the base to use for a [LinearScale](#), which is logarithmic. If [LinearScale.Logarithmic](#) is set to true, the **LinearScale.LogarithmicBase** element is ignored.

The **LinearScale.LogarithmicBase** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 10. This element MUST have a **Float** value that is greater than 1.

The following is the parent element of the **LinearScale.LogarithmicBase** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.LogarithmicBase** element.

```
<xsd:element name="LogarithmicBase" type="xsd:string" minOccurs="0" />
```

## 2.175.12 LinearScale.MaximumPin

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearScale.MaximumPin** element specifies the maximum value at which a [LinearPointer](#) on a [LinearScale](#) stops. This element is optional. This element is of type [ScalePin](#).

The following is the parent element of the **LinearScale.MaximumPin** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.MaximumPin** element.

```
<xsd:element name="MaximumPin" type="ScalePinType" minOccurs="0" />
```

## 2.175.13 LinearScale.MaximumValue

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearScale.MaximumValue** element specifies the maximum value for a [LinearScale](#). This element is optional.

If this element is present, the value of [GaugeInputValue.Value](#) within the **LinearScale.MaximumValue** element MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a [Float](#). If **GaugeInputValue.Value** is not set within the **LinearScale.MaximumValue** element, the value of **LinearScale.MaximumValue** is interpreted as 100. The **GaugeInputValue.Value** within this element MUST be greater than the **GaugeInputValue.Value** within its peer [LinearScale.MinimumValue](#) element.

The **LinearScale.MaximumValue** element is of type [GaugeInputValue](#).

The following is the parent element of the **LinearScale.MaximumValue** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.MaximumValue** element.

```
<xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
```

## 2.175.14 LinearScale.MinimumPin

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearScale.MinimumPin** element specifies the minimum value at which a [LinearPointer](#) on a [LinearScale](#) stops. This element is optional. This element is of type [ScalePin](#).

The following is the parent element of the **LinearScale.MinimumPin** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.MinimumPin** element.

```
<xsd:element name="MinimumPin" type="ScalePinType" minOccurs="0" />
```

### 2.175.15 LinearScale.MinimumValue

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScale.MinimumValue** element specifies the minimum value for a [LinearScale](#). This element is optional.

If this element is present, the [GaugeInputValue.Value](#) within the **LinearScale.MinimumValue** element MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. If **GaugeInputValue.Value** is not set within the **LinearScale.MinimumValue** element, the value of **LinearScale.MinimumValue** is interpreted as 0. The **GaugeInputValue.Value** within the **LinearScale.MinimumValue** element MUST be less than the **GaugeInputValue.Value** within its peer [LinearScale.MaximumValue](#) element.

The **LinearScale.MinimumValue** element is of type [GaugeInputValue](#).

The following is the parent element of the **LinearScale.MinimumValue** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.MinimumValue** element.

```
<xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
```

### 2.175.16 LinearScale.Multiplier

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScale.Multiplier** element specifies the multiplication values for a [LinearScale](#) before they are displayed. This element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 1.

The following is the parent element of the **LinearScale.Multiplier** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.Multiplier** element.

```
<xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
```

## 2.175.17 LinearScale.Reversed

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScale.Reversed** element specifies whether the direction of a [LinearScale](#) is reversed. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **LinearScale.Reversed** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.Reversed** element.

```
<xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
```

## 2.175.18 LinearScale.ScaleLabels

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScale.ScaleLabels** element specifies labels to display on a [LinearScale](#). This element is of type [ScaleLabels](#). This element is optional.

The following is the parent element of the **LinearScale.ScaleLabels** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.ScaleLabels** element.

```
<xsd:element name="ScaleLabels" type="ScaleLabelsType" minOccurs="0" />
```

## 2.175.19 LinearScale.ScaleRanges

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScale.ScaleRanges** element specifies ranges to display on a [LinearScale](#). This element is optional. This element is of type [ScaleRanges](#).

The following is the parent element of the **LinearScale.ScaleRanges** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.ScaleRanges** element.

```
<xsd:element name="ScaleRanges" type="ScaleRangesType" minOccurs="0" />
```

## 2.175.20 LinearScale.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScale.Style** element specifies style properties for a [LinearScale](#). The **LinearScale.Style** element is optional. This element is of type [Style](#).

The following is the parent element of the **LinearScale.Style** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.175.21 LinearScale.TickMarksOnTop

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScale.TickMarksOnTop** element specifies whether [GaugeTickMarks](#) within a [LinearScale](#) is drawn atop [LinearPointer](#) instances within [LinearScale.GaugePointers](#).

The **LinearScale.TickMarksOnTop** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **LinearScale.TickMarksOnTop** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.TickMarksOnTop** element.

```
<xsd:element name="TickMarksOnTop" type="xsd:string" minOccurs="0" />
```

## 2.175.22 LinearScale.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScale.ToolTip** element specifies the tooltip text for a [LinearScale](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **LinearScale.ToolTip** element is optional. The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **LinearScale.ToolTip** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

### 2.175.23 LinearScale.Width

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearScale.Width** element specifies the width for a [LinearScale](#) as a percentage of [LinearGauge](#) size. The **LinearScale.Width** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **LinearScale.Width** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
```

### 2.175.24 LinearScale.EndMargin

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearScale.EndMargin** element specifies the distance between the end of a [LinearScale](#) and its parent [LinearGauge](#) distance, as a percentage of the size of the **LinearGauge**. The **LinearScale.EndMargin** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0 and less than or equal to 100 (specifying the percentage of the container). If this element is not present, its value is interpreted as 8.

The following is the parent element of the **LinearScale.EndMargin** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.EndMargin** element.

```
<xsd:element name="EndMargin" type="xsd:string" minOccurs="0">
```

### 2.175.25 LinearScale.Position

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearScale.Position** element specifies the position of a [LinearScale](#) as a percentage of the size of the [LinearGauge](#), (or the [LinearGauge.Height](#) if [LinearGauge.Orientation](#) is set to "vertical", or the [LinearGauge.Width](#) if [LinearGauge.Orientation](#) is set to "horizontal").

The **LinearScale.Position** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 50.

The following is the parent element of the **LinearScale.Position** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.Position** element.

```
<xsd:element name="Position" type="xsd:string" minOccurs="0">
```

## 2.175.26 LinearScale.StartMargin

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearScale.StartMargin** element specifies the distance between the start of a [LinearScale](#) and its parent [LinearGauge](#) distance, as a percentage of the size of the **LinearGauge**. The **LinearScale.StartMargin** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0 and less than or equal to 100 (specifying the percentage of the container). If this element is not present, its value is interpreted as 8.

The following is the parent element of the **LinearScale.StartMargin** element.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.StartMargin** element.

```
<xsd:element name="StartMargin" type="xsd:string" minOccurs="0">
```

## 2.176 CustomLabels

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomLabels** element specifies a set of custom labels for a [RadialScale](#) or a [LinearScale](#). The **CustomLabels** element MUST contain at least one [CustomLabels.CustomLabel](#) instance.

The following are the parent and child elements of the **CustomLabels** element.

Parent elements
LinearScale
RadialScale

Child elements
CustomLabels.CustomLabel

The following is the XML Schema definition of the **CustomLabels** element in RDL 2008/01.

```
<xsd:complexType name="CustomLabelsType">
```



```

<xsd:sequence>
  <xsd:element name="CustomLabel" type="CustomLabelType" minOccurs="1"
    maxOccurs="unbounded" />
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **CustomLabels** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="CustomLabelsType">
  <xsd:sequence>
    <xsd:element name="CustomLabel" type="CustomLabelType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.176.1 CustomLabels.CustomLabel

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomLabels.CustomLabel** element specifies a custom label to be placed on a [RadialScale](#) or [LinearScale](#) element. The **CustomLabels.CustomLabel** element is of type [CustomLabel](#). The **CustomLabels.CustomLabel** element MUST be specified at least once within a [CustomLabels](#) collection.

The following is the parent element of the **CustomLabels.CustomLabel** element.

Parent elements
CustomLabels

The following is the XML Schema definition of the **CustomLabels.CustomLabel** element.

```

<xsd:element name="CustomLabel" type="CustomLabelType" minOccurs="1"
  maxOccurs="unbounded" />

```

### 2.177 CustomLabel

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomLabel** element specifies a custom label for a [RadialScale](#) or a [LinearScale](#) element. The **CustomLabel** element MUST be specified at least once within a [CustomLabels](#) collection.

The following are the parent elements, attributes, and child elements of the **CustomLabel** element.

Parent elements
CustomLabels

Attributes
<a href="#">CustomLabel.Name</a>

Child elements
<a href="#">CustomLabel.AllowUpsideDown</a>
<a href="#">CustomLabel.DistanceFromScale</a>
<a href="#">CustomLabel.FontAngle</a>
<a href="#">CustomLabel.Hidden</a>
<a href="#">CustomLabel.Placement</a>
<a href="#">CustomLabel.RotateLabel</a>
<a href="#">CustomLabel.Style</a>
<a href="#">CustomLabel.Text</a>
<a href="#">CustomLabel.TickMarkStyle</a>
<a href="#">CustomLabel.UseFontPercent</a>
<a href="#">CustomLabel.Value</a>

The following is the XML Schema definition of the **CustomLabel** element in RDL 2008/01.

```
<xsd:complexType name="CustomLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkStyle" type="TickMarkStyleType" minOccurs="0" />
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="RotateLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **CustomLabel** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="CustomLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkStyle" type="TickMarkStyleType" minOccurs="0" />
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="RotateLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
</xsd:complexType>
```

```

</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.177.1 CustomLabel.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomLabel.Name** attribute specifies a unique identifier for a [CustomLabel](#). The **CustomLabel.Name** attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [UTR15] that is unique within a particular [CustomLabels](#) collection.

The following is the parent element of the **CustomLabel.Name** attribute.

Parent elements
CustomLabel

The following is the XML Schema definition of the **CustomLabel.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.177.2 CustomLabel.AllowUpsideDown

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomLabel.AllowUpsideDown** element specifies whether a [CustomLabel](#) can be rotated by more than 90 degrees. The **CustomLabel.AllowUpsideDown** element is optional.

If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false. If the containing element for this **CustomLabel** is not a [RadialScale](#), the **CustomLabel.AllowUpsideDown** element is ignored.

The following is the parent element of the **CustomLabel.AllowUpsideDown** element.

Parent elements
CustomLabel

The following is the XML Schema definition of the **CustomLabel.AllowUpsideDown** element.

```
<xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
```

### 2.177.3 CustomLabel.DistanceFromScale

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomLabel.DistanceFromScale** element specifies the distance from a [CustomLabel](#) to its containing element as a percentage of the containing element size, as specified by radius in a [RadialScale](#) and by length in a [LinearScale](#). The **CustomLabel.DistanceFromScale** element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**.[<34>](#) If this element is not present, its value is interpreted as 0.

Following is the parent element of the **CustomLabel.DistanceFromScale** element.

Parent elements
CustomLabel

The following is the XML Schema definition of the **CustomLabel.DistanceFromScale** element.

```
<xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
```

#### 2.177.4 CustomLabel.FontAngle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomLabel.FontAngle** element specifies the angle of rotation, in degrees, for the text of a [CustomLabel](#) relative to its position within its containing [RadialScale](#) or [LinearScale](#) element. The **CustomLabel.FontAngle** element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0 and less than or equal to 360. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **CustomLabel.FontAngle** element.

Parent elements
CustomLabel

The following is the XML Schema definition of the **CustomLabel.FontAngle** element.

```
<xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
```

#### 2.177.5 CustomLabel.Hidden

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomLabel.Hidden** element specifies whether a [CustomLabel](#) is hidden. The **CustomLabel.Hidden** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **CustomLabel.Hidden** element.

Parent elements
CustomLabel

The following is the XML Schema definition of the **CustomLabel.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

## 2.177.6 CustomLabel.Placement

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomLabel.Placement** element specifies where a [CustomLabel](#) is placed relative to its containing [RadialScale](#) or [LinearScale](#) element. The **CustomLabel.Placement** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Inside:** Specifies that the **CustomLabel** instance is placed inside of the **RadialScale** or **LinearScale** element.

**Outside:** Specifies that the **CustomLabel** instance is placed outside of the **RadialScale** or **LinearScale** element.

**Cross:** Specifies that the **CustomLabel** instance is placed across the **RadialScale** or **LinearScale** element.

If this element is not present, its value is interpreted as "Inside".

The following is the parent element of the **CustomLabel.Placement** element.

Parent elements
CustomLabel

The following is the XML Schema definition of the **CustomLabel.Placement** element.

```
<xsd:element name="Placement" type="xsd:string" minOccurs="0" />
```

## 2.177.7 CustomLabel.RotateLabel

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomLabel.RotateLabel** element specifies whether the text for a [CustomLabel](#) rotates along with its containing element. The **CustomLabel.RotateLabel** element is optional.

If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false. If the containing element is not a [RadialScale](#), the **CustomLabel.RotateLabel** element is ignored.

The following is the parent element of the **CustomLabel.RotateLabel** element.

Parent elements
CustomLabel

The following is the XML Schema definition of the **CustomLabel.RotateLabel** element.

```
<xsd:element name="RotateLabel" type="xsd:string" minOccurs="0" />
```

## 2.177.8 CustomLabel.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomLabel.Style** element specifies style properties for a [CustomLabel](#). The **CustomLabel.Style** element is of type [Style](#). The **CustomLabel.Style** element is optional.

The following is the parent element of the **CustomLabel.Style** element.

Parent elements
CustomLabel

The following is the XML Schema definition of the **CustomLabel.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.177.9 CustomLabel.Text

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomLabel.Text** element specifies the text for a [CustomLabel](#). The **CustomLabel.Text** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **CustomLabel.Text** element.

Parent elements
CustomLabel

The following is the XML Schema definition of the **CustomLabel.Text** element.

```
<xsd:element name="Text" type="xsd:string" minOccurs="0" />
```

## 2.177.10 CustomLabel.TickMarkStyle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomLabel.TickMarkStyle** element specifies style properties for the tick mark to which a [CustomLabel](#) instance is attached. The attached tick mark is specified by the peer [CustomLabel.Value](#) element. The **CustomLabel.TickMarkStyle** element is optional and is of type [TickMarkStyle](#).

The following is the parent element of the **CustomLabel.TickMarkStyle** element.

Parent elements
CustomLabel

The following is the XML Schema definition of the **CustomLabel.TickMarkStyle** element.

```
<xsd:element name="TickMarkStyle" type="TickMarkStyleType" minOccurs="0" />
```

## 2.177.11 CustomLabel.UseFontPercent

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomLabel.UseFontPercent** element specifies whether the font size for a [CustomLabel](#) is measured as a percentage of the parent element or in the units specified by the [Style.FontSize](#) element of the [CustomLabel.Style](#) element. The **CustomLabel.UseFontPercent** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **CustomLabel.UseFontPercent** element.

Parent elements
<a href="#">CustomLabel</a>

The following is the XML Schema definition of the **CustomLabel.UseFontPercent** element.

```
<xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
```

## 2.177.12 CustomLabel.Value

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomLabel.Value** element specifies the position on a [RadialScale](#) or [LinearScale](#) element where a [CustomLabel](#) will be placed. The **CustomLabel.Value** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **CustomLabel.Value** element.

Parent elements
<a href="#">CustomLabel</a>

The following is the XML Schema definition of the **CustomLabel.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="0" />
```

## 2.178 TickMarkStyle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TickMarkStyle** element specifies appearance properties for a tick mark that is associated with a [CustomLabel](#) instance. The **TickMarkStyle** element is optional.

The following are the parent and child elements of the **TickMarkStyle** element.

Parent elements
<a href="#">CustomLabel</a>

Child elements
<a href="#">TickMarkStyle.DistanceFromScale</a>

Child elements
<a href="#">TickMarkStyle.EnableGradient</a>
<a href="#">TickMarkStyle.GradientDensity</a>
<a href="#">TickMarkStyle.Hidden</a>
<a href="#">TickMarkStyle.Length</a>
<a href="#">TickMarkStyle.Placement</a>
<a href="#">TickMarkStyle.Shape</a>
<a href="#">TickMarkStyle.Style</a>
<a href="#">TickMarkStyle.TickMarkImage</a>
<a href="#">TickMarkStyle.Width</a>

The following is the XML Schema definition of the **TickMarkStyle** element in RDL 2008/01.

```
<xsd:complexType name="TickMarkStyleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
    <xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shape" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TickMarkStyle** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="TickMarkStyleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
    <xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shape" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.178.1 TickMarkStyle.DistanceFromScale

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)



The **TickMarkStyle.DistanceFromScale** element specifies the distance from the tick mark that is associated with a [CustomLabel](#) instance to a containing [RadialScale](#) or [LinearScale](#). The **TickMarkStyle.DistanceFromScale** element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**.<35> If this element is not present, its value is interpreted as 0.

The following is the parent element of the **TickMarkStyle.DistanceFromScale** element.

Parent elements
<a href="#">TickMarkStyle</a>

The following is the XML Schema definition of the **TickMarkStyle.DistanceFromScale** element.

```
<xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0">
```

### 2.178.2 TickMarkStyle.EnableGradient

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TickMarkStyle.EnableGradient** element specifies whether a gradient effect is used for the tick mark for a [CustomLabel](#) instance. The **TickMarkStyle.EnableGradient** element is optional.

If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **TickMarkStyle.EnableGradient** element.

Parent elements
<a href="#">TickMarkStyle</a>

The following is the XML Schema definition of the **TickMarkStyle.EnableGradient** element.

```
<xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
```

### 2.178.3 TickMarkStyle.GradientDensity

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TickMarkStyle.GradientDensity** element specifies the intensity of the gradient effect for the tick mark on a [CustomLabel](#). This element is optional.

If the **TickMarkStyle.GradientDensity** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0 and less than or equal to 100, specifying the percentage of maximum intensity. If this element is not present, its value is interpreted as 30.

The following is the parent element of the **TickMarkStyle.GradientDensity** element.

Parent elements
<a href="#">TickMarkStyle</a>

The following is the XML Schema definition of the **TickMarkStyle.GradientDensity** element.

```
<xsd:element name="GradientDensity" type="xsd:string" minOccurs="0">
```

## 2.178.4 TickMarkStyle.Hidden

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TickMarkStyle.Hidden** element specifies whether the tick mark within a [CustomLabel](#) instance is hidden. The **TickMarkStyle.Hidden** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **TickMarkStyle.Hidden** element.

Parent elements
<a href="#">TickMarkStyle</a>

The following is the XML Schema definition of the **TickMarkStyle.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0">
```

## 2.178.5 TickMarkStyle.Length

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TickMarkStyle.Length** element specifies the length of the tick mark for a [CustomLabel](#) as a percentage of the radius of a containing [RadialScale](#) or the minimum of the length and width of a containing [LinearScale](#). This element is optional.

If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0. [<36>](#) If this element is not present, its value is interpreted as 0.

The following is the parent element of the **TickMarkStyle.Length** element.

Parent elements
<a href="#">TickMarkStyle</a>

The following is the XML Schema definition of the **TickMarkStyle.Length** element.

```
<xsd:element name="Length" type="xsd:string" minOccurs="0">
```

## 2.178.6 TickMarkStyle.Placement

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TickMarkStyle.Placement** element specifies where the tick mark for a [CustomLabel](#) is placed relative to its associated [RadialScale](#) or [LinearScale](#). This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Inside:** The tick mark associated with a **CustomLabel** instance is placed inside the [GaugeTickMarks](#) instances in the **radial scale** or **linear scale**.

**Outside:** The tick mark associated with a **CustomLabel** instance is placed outside the **GaugeTickMarks** instances within the radial scale or linear scale.

**Cross:** The tick mark associated with a **CustomLabel** instance is across the **GaugeTickMarks** instances within the radial scale or linear scale.

If the **TickMarkStyle.Placement** element is not present, its value is interpreted as "Inside".

The following is the parent element of the **TickMarkStyle.Placement** element.

Parent elements
<a href="#">TickMarkStyle</a>

The following is the XML Schema definition of the **TickMarkStyle.Placement** element.

```
<xsd:element name="Placement" type="xsd:string" minOccurs="0">
```

## 2.178.7 TickMarkStyle.Shape

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TickMarkStyle.Shape** element specifies the shape of a tick mark for a [CustomLabel](#) instance. This element is optional. If the **TickMarkStyle.Shape** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Rectangle:** The marker is of type "Rectangle".

**Triangle:** The marker is of type "Triangle".

**Circle:** The marker is of type "Circle".

**Diamond:** The marker is of type "Diamond".

**Trapezoid:** The marker is of type "Trapezoid".

**Star:** The marker is of type "Star".

**Wedge:** The marker is of type "Wedge".

**Pentagon:** The marker is of type "Pentagon".

**None:** The marker is of type "None".

If the **TickMarkStyle.Shape** element is not present, its value is interpreted as "Rectangle".

The following is the parent element of the **TickMarkStyle.Shape** element.

Parent elements
<a href="#">TickMarkStyle</a>

The following is the XML Schema definition of the **TickMarkStyle.Shape** element.

```
<xsd:element name="Shape" type="xsd:string" minOccurs="0">
```

## 2.178.8 TickMarkStyle.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TickMarkStyle.Style** element specifies style properties for the tick mark that is associated with a [CustomLabel](#). The **TickMarkStyle.Style** element is optional. This element is of type [Style](#).

The following is the parent element of the **TickMarkStyle.Style** element.

Parent elements
<a href="#">TickMarkStyle</a>

The following is the XML Schema definition of the **TickMarkStyle.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0">
```

## 2.178.9 TickMarkStyle.TickMarkImage

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TickMarkStyle.TickMarkImage** element specifies the image to use for the tick mark that is associated with a [CustomLabel](#). The **TickMarkStyle.TickMarkImage** element is optional. This element is of type [TopImage](#).

The following is the parent element of the **TickMarkStyle.TickMarkImage** element.

Parent elements
<a href="#">TickMarkStyle</a>

The following is the XML Schema definition of the **TickMarkStyle.TickMarkImage** element.

```
<xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0">
```

## 2.178.10 TickMarkStyle.Width

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TickMarkStyle.Width** element specifies the width of the tick mark for a [CustomLabel](#) as a percentage of the radius of a containing [RadialScale](#) or the minimum of the length and width of a containing [LinearScale](#). This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0. [<37>](#) If this element is not present, its value is interpreted as "0".

The following is the parent element of the **TickMarkStyle.Width** element.

Parent elements
<a href="#">TickMarkStyle</a>

The following is the XML Schema definition of the **TickMarkStyle.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0">
```

## 2.179 GaugeInputValue

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeInputValue** element specifies an expression and an optional formula used within a [LinearGauge](#) or a [RadialGauge](#).

The following are the parent and child elements of the **GaugeInputValue** element.

Parent elements
<a href="#">LinearScale</a>
<a href="#">LinearPointer</a>
<a href="#">ScaleRange</a>
<a href="#">NumericIndicator</a>
<a href="#">NumericIndicatorRange</a>
<a href="#">RadialScale</a>
<a href="#">RadialPointer</a>
<a href="#">StateIndicator</a>
<a href="#">IndicatorState</a>

Child elements
<a href="#">GaugeInputValue.AddConstant</a>
<a href="#">GaugeInputValue.DataElementName</a>
<a href="#">GaugeInputValue.DataElementOutput</a>
<a href="#">GaugeInputValue.Formula</a>
<a href="#">GaugeInputValue.MaxPercent</a>
<a href="#">GaugeInputValue.MinPercent</a>
<a href="#">GaugeInputValue.Multiplier</a>
<a href="#">GaugeInputValue.Value</a>

The following is the XML Schema definition of the **GaugeInputValue** element in RDL 2008/01.

```
<xsd:complexType name="GaugeInputValueType">  
  <xsd:choice maxOccurs="unbounded">  
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />  
    <xsd:element name="Formula" type="xsd:string" minOccurs="0" />  
    <xsd:element name="MinPercent" type="xsd:string" minOccurs="0" />  
    <xsd:element name="MaxPercent" type="xsd:string" minOccurs="0" />  
  </xsd:choice>  
</xsd:complexType>
```

```

<xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
<xsd:element name="AddConstant" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **GaugeInputValue** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="GaugeInputValueType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="Formula" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinPercent" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaxPercent" type="xsd:string" minOccurs="0" />
    <xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
    <xsd:element name="AddConstant" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

## 2.179.1 GaugeInputValue.AddConstant

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeInputValue.AddConstant** element specifies the constant to add to the value of [GaugeInputValue.Value](#) after multiplying by the value of [GaugeInputValue.Multiplier](#).

The **GaugeInputValue.AddConstant** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0. If the peer element [GaugeInputValue.Formula](#) is not set to "Linear", the **GaugeInputValue.AddConstant** element is ignored.

The following is the parent element of the **GaugeInputValue.AddConstant** element.

<b>Parent elements</b>
<a href="#">GaugeInputValue</a>

The following is the XML Schema definition of the **GaugeInputValue.AddConstant** element.

```
<xsd:element name="AddConstant" type="xsd:string" minOccurs="0">
```

## 2.179.2 GaugeInputValue.DataElementName

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeInputValue.DataElementName** element specifies the name to use for the data element or attribute for [GaugeInputValue](#) in a **data rendering**. The **GaugeInputValue.DataElementName** element is optional. This element MUST be a **CLS-compliant identifier** [UTR15]. The value of this element is interpreted as a [String](#) ([XMLSCHEMA2/2] section 3.2.1).

The following is the parent element of the **GaugeInputValue.DataElementName** element.

Parent elements
GaugeInputValue

The following is the XML Schema definition of the **GaugeInputValue.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0">
```

## 2.179.3 GaugeInputValue.DataElementOutput

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeInputValue.DataElementOutput** element indicates whether the parent element of the [GaugeInputValue](#) appears in a **data rendering**. The **GaugeInputValue.DataElementOutput** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) constant that is one of the following:

**Output:** Specifies that the item appears in the data rendering output.

**NoOutput:** Specifies that the item does not appear in the data rendering output.

If this element is not present, its value is interpreted as "Output".

The following is the parent element of the **GaugeInputValue.DataElementOutput** element.

Parent elements
GaugeInputValue

The following is the XML Schema definition of the **GaugeInputValue.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.179.4 GaugeInputValue.Formula

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeInputValue.Formula** element specifies the type of calculation to perform on the values represented by [GaugeInputValue.Value](#), if more than one value is present.

The **GaugeInputValue.Formula** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. This **String** value MUST be one of the following:

**None:** Specifies that the last value is used in the sorted array of values.

**Average:** Specifies that the average of the values is used.

**Linear:** Specifies that a linear formula is applied to the last value in the set of sorted values.

**Max:** Specifies that the maximum value is used.

**Min:** Specifies that a minimum value is used.

**Median:** Specifies that the median value is used.

**OpenClose:** Specifies that the minimum and maximum values are used.

**Percentile:** Specifies that two values are used that respectively represent a lower percentile and a higher percentile.

**Variance:** Specifies that the variance of the values is used.

**RateOfChange:** Specifies that the rate of change of the values is used over a specified time interval.

**Integral:** Specifies that the integral of the values is calculated over a specified time interval.

If the **GaugeInputValue.Formula** element is not present, its value is interpreted as "None".

The following is the parent element of the **GaugeInputValue.Formula** element.

Parent elements
<a href="#">GaugeInputValue</a>

The following is the XML Schema definition of the **GaugeInputValue.Formula** element.

```
<xsd:element name="Formula" type="xsd:string" minOccurs="0">
```

## 2.179.5 GaugeInputValue.MaxPercent

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeInputValue.MaxPercent** element specifies the maximum percentage to use for one of the two values returned for [GaugeInputValue.Value](#) if the peer element [GaugeInputValue.Formula](#) is set to "Percentile".

The **GaugeInputValue.MaxPercent** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. This element is ignored if peer element **GaugeInputValue.Formula** is not set to "Percentile". The value of the **GaugeInputValue.MaxPercent** element MUST be non-negative. If this element is not present, its value is interpreted as "NaN" (not a number).



The following is the parent element of the **GaugeInputValue.MaxPercent** element.

<b>Parent elements</b>
<a href="#">GaugeInputValue</a>

The following is the XML Schema definition of the **GaugeInputValue.MaxPercent** element.

```
<xsd:element name="MaxPercent" type="xsd:string" minOccurs="0">
```

### 2.179.6 GaugeInputValue.MinPercent

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeInputValue.MinPercent** element specifies the minimum percentage to use for one of the two values returned for [GaugeInputValue.Value](#) if [GaugeInputValue.Formula](#) is set to "Percentile".

The **GaugeInputValue.MinPercent** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. This element is ignored if peer element **GaugeInputValue.Formula** is not set to "Percentile". The value of the **GaugeInputValue.MinPercent** element MUST be non-negative. If this element is not present, its value is interpreted as "NaN".

The following is the parent element of the **GaugeInputValue.MinPercent** element.

<b>Parent elements</b>
<a href="#">GaugeInputValue</a>

The following is the XML Schema definition of the **GaugeInputValue.MinPercent** element.

```
<xsd:element name="MinPercent" type="xsd:string" minOccurs="0">
```

### 2.179.7 GaugeInputValue.Multiplier

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeInputValue.Multiplier** element specifies the value to multiply to the value of [GaugeInputValue.Value](#). The **GaugeInputValue.Multiplier** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 1. If the peer element [GaugeInputValue.Formula](#) is not set to "Linear", this element is ignored.

The following is the parent element of the **GaugeInputValue.Multiplier** element.

<b>Parent elements</b>
<a href="#">GaugeInputValue</a>

The following is the XML Schema definition of the **GaugeInputValue.Multiplier** element.

```
<xsd:element name="Multiplier" type="xsd:string" minOccurs="0">
```

## 2.179.8 GaugeInputValue.Value

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeInputValue.Value** element specifies an expression for the value of a [GaugeInputValue](#) instance. The **GaugeInputValue.Value** element MUST be specified. The value of the **GaugeInputValue.Value** element MUST be a **Numeric** or an expression that evaluates to a **Numeric**.

The following is the parent element of the **GaugeInputValue.Value** element.

Parent elements
<a href="#">GaugeInputValue</a>

The following is the XML Schema definition of the **GaugeInputValue.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1">
```

## 2.180 GaugeTickMarks

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeTickMarks** element specifies tick marks along a [LinearScale](#) or [RadialScale](#). The **GaugeTickMarks** element MUST be represented by one of the following:

- [LinearScale.GaugeMajorTickMarks](#)
- [LinearScale.GaugeMinorTickMarks](#)
- [RadialScale.GaugeMajorTickMarks](#)
- [RadialScale.GaugeMinorTickMarks](#)

The following are the parent and child elements of the **GaugeTickMarks** element.

Parent elements
<a href="#">LinearScale</a>
<a href="#">RadialScale</a>

Child elements
<a href="#">GaugeTickMarks.Interval</a>
<a href="#">GaugeTickMarks.IntervalOffset</a>
<a href="#">GaugeTickMarks.DistanceFromScale</a>
<a href="#">GaugeTickMarks.EnableGradient</a>
<a href="#">GaugeTickMarks.GradientDensity</a>
<a href="#">GaugeTickMarks.Hidden</a>
<a href="#">GaugeTickMarks.Length</a>

Child elements
<a href="#">GaugeTickMarks.Placement</a>
<a href="#">GaugeTickMarks.Shape</a>
<a href="#">GaugeTickMarks.Style</a>
<a href="#">GaugeTickMarks.TickMarkImage</a>
<a href="#">GaugeTickMarks.Width</a>

The following is the XML Schema definition of the **GaugeTickMarks** element in RDL 2008/01.

```
<xsd:complexType name="GaugeTickMarksType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--TickMarkStyleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
    <xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shape" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <!--TickMarkStyleTypeEnd-->
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **GaugeTickMarks** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="GaugeTickMarksType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--TickMarkStyleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
    <xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shape" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <!--TickMarkStyleTypeEnd-->
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.180.1 GaugeTickMarks.Interval

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeTickMarks.Interval** element defines the interval between tick marks within a [GaugeTickMarks](#) element. The **GaugeTickMarks.Interval** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its **Float** value is interpreted as that of the [RadialScale.Interval](#) or [LinearScale.Interval](#) value for the respective parent [RadialScale](#) or [LinearScale](#) element. The **Float** value of the **GaugeTickMarks.Interval** element MUST be non-negative.

The following is the parent element of the **GaugeTickMarks.Interval** element.

Parent elements
<a href="#">GaugeTickMarks</a>

The following is the XML Schema definition of the **GaugeTickMarks.Interval** element.

```
<xsd:element name="Interval" type="xsd:string" minOccurs="0">
```

### 2.180.2 GaugeTickMarks.IntervalOffset

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeTickMarks.IntervalOffset** element defines the offset for the first tick mark within a [GaugeTickMarks](#) element. The **GaugeTickMarks.IntervalOffset** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its **Float** value is interpreted as that of the [RadialScale.IntervalOffset](#) or [LinearScale.IntervalOffset](#) value for the respective parent [RadialScale](#) or [LinearScale](#) element. The **Float** value of the **GaugeTickMarks.IntervalOffset** element MUST be non-negative.

The following is the parent element of the **GaugeTickMarks.IntervalOffset** element.

Parent elements
<a href="#">GaugeTickMarks</a>

The following is the XML Schema definition of the **GaugeTickMarks.IntervalOffset** element.

```
<xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0">
```

### 2.180.3 GaugeTickMarks.DistanceFromScale

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **GaugeTickMarks.DistanceFromScale** element specifies the distance from the tick marks that is associated with a label to a containing [LinearScale](#) or [RadialScale](#). The **GaugeTickMarks.DistanceFromScale** element is optional.

If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**.<38> If this element is not present, its value is interpreted as 0.

Following is the parent element of the **GaugeTickMarks.DistanceFromScale** element.

Parent elements
<a href="#">GaugeTickMarks</a>

The following is the XML Schema definition of the **GaugeTickMarks.DistanceFromScale** element.

```
<xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0">
```

#### 2.180.4 GaugeTickMarks.EnableGradient

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeTickMarks.EnableGradient** element specifies whether a gradient effect is used for tick marks. This element is optional.

If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **GaugeTickMarks.EnableGradient** element.

Parent elements
<a href="#">GaugeTickMarks</a>

The following is the XML Schema definition of the **GaugeTickMarks.EnableGradient** element.

```
<xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
```

#### 2.180.5 GaugeTickMarks.GradientDensity

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeTickMarks.GradientDensity** element specifies the intensity of the gradient effect for tick marks. This element is optional.

If the **GaugeTickMarks.GradientDensity** element is present, its value MUST be a [Float](#) [IEEE754] ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0 and less than or equal to 100, specifying the percentage of maximum gradient density. If this element is not present, its value is interpreted as 30.

The following is the parent element of the **GaugeTickMarks.GradientDensity** element.

Parent elements
<a href="#">GaugeTickMarks</a>

The following is the XML Schema definition of the **GaugeTickMarks.GradientDensity** element.

```
xsd:element name="GradientDensity" type="xsd:string" minOccurs="0">
```

#### 2.180.6 GaugeTickMarks.Hidden

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeTickMarks.Hidden** element specifies whether the tick marks are hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or

an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **GaugeTickMarks.Hidden** element.

Parent elements
<a href="#">GaugeTickMarks</a>

The following is the XML Schema definition of the **GaugeTickMarks.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0">
```

## 2.180.7 GaugeTickMarks.Length

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeTickMarks.Length** element specifies the length of the tick marks as a percentage of the radius of a containing [RadialScale](#) or the minimum of the length and width of a containing [LinearScale](#). This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0. [<39>](#) If this element is not present, its value is interpreted as 0.

Following is the parent element of the **GaugeTickMarks.Length** element.

Parent elements
<a href="#">GaugeTickMarks</a>

The following is the XML Schema definition of the **GaugeTickMarks.Length** element.

```
<xsd:element name="Length" type="xsd:string" minOccurs="0">
```

## 2.180.8 GaugeTickMarks.Placement

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeTickMarks.Placement** element specifies where a tick mark is placed relative to its associated [LinearScale](#) or [RadialScale](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Inside:** The tick mark is placed inside the [GaugeTickMarks](#) instances in the **radial scale** or **linear scale**.

**Outside:** The tick mark is placed outside the **GaugeTickMarks** instances within the radial scale or linear scale.

**Cross:** The tick mark is placed across the **GaugeTickMarks** instances within the radial scale or linear scale.

If the **GaugeTickMarks.Placement** element is not present, its value is interpreted as "Inside".

Following is the parent element of the **GaugeTickMarks.Placement** element.

Parent elements
<a href="#">GaugeTickMarks</a>

The following is the XML Schema definition of the **GaugeTickMarks.Placement** element.

```
<xsd:element name="Placement" type="xsd:string" minOccurs="0">
```

### 2.180.9 GaugeTickMarks.Shape

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeTickMarks.Shape** element specifies the shape of a tick mark. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Rectangle:** The marker is of type "Rectangle".

**Triangle:** The marker is of type "Triangle".

**Circle:** The marker is of type "Circle".

**Diamond:** The marker is of type "Diamond".

**Trapezoid:** The marker is of type "Trapezoid".

**Star:** The marker is of type "Star".

**Wedge:** The marker is of type "Wedge".

**Pentagon:** The marker is of type "Pentagon".

**None:** The marker is of type "None".

If the **GaugeTickMarks.Shape** element is not present, its value is interpreted as "Rectangle".

Following is the parent element of the **GaugeTickMarks.Shape** element.

Parent elements
<a href="#">GaugeTickMarks</a>

The following is the XML Schema definition of the **GaugeTickMarks.Shape** element.

```
<xsd:element name="Shape" type="xsd:string" minOccurs="0">
```

### 2.180.10 GaugeTickMarks.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeTickMarks.Style** element specifies style properties for a tick mark. This element is optional and is of type [Style](#).

Following is the parent element of the **GaugeTickMarks.Style** element.

Parent elements
<a href="#">GaugeTickMarks</a>

The following is the XML Schema definition of the **GaugeTickMarks.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0">
```

### 2.180.11 GaugeTickMarks.TickMarkImage

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeTickMarks.TickMarkImage** element specifies the image to use for a tick mark. This element is optional. This element is of type [TopImage](#).

Following is the parent element of the **GaugeTickMarks.TickMarkImage** element.

Parent elements
<a href="#">GaugeTickMarks</a>

The following is the XML Schema definition of the **GaugeTickMarks.TickMarkImage** element.

```
<xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0">
```

### 2.180.12 GaugeTickMarks.Width

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **GaugeTickMarks.Width** element specifies the width of a tick mark as a percentage of the radius of a containing [RadialScale](#) or the minimum of the length and width of a containing [LinearScale](#). This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0. [<40>](#) If this element is not present, its value is interpreted as 0.

Following is the parent element of the **GaugeTickMarks.Width** element.

Parent elements
<a href="#">GaugeTickMarks</a>

The following is the XML Schema definition of the **GaugeTickMarks.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0">
```

### 2.181 LinearPointers

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)



The **LinearPointers** element specifies the collection of [LinearPointer](#) instances for a [LinearScale](#). The **LinearPointers** element is optional. If this element is present, it MUST contain at least one [LinearPointers.LinearPointer](#) instance.

The following are the parent and child elements of the **LinearPointers** element.

Parent elements
LinearScale

Child elements
LinearPointers.LinearPointer

The following is the XML Schema definition of the **LinearPointers** element in RDL 2008/01.

```
<xsd:complexType name="LinearPointersType">
  <xsd:sequence>
    <xsd:element name="LinearPointer" type="LinearPointerType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **LinearPointers** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="LinearPointersType">
  <xsd:sequence>
    <xsd:element name="LinearPointer" type="LinearPointerType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.181.1 LinearPointers.LinearPointer

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearPointers.LinearPointer** element specifies a [LinearPointer](#) within the collection of [LinearPointers](#) for a [LinearScale](#). This element MUST be specified at least once within a **LinearPointers** collection. This element is of type **LinearPointer**.

The following is the parent element of the **LinearPointers.LinearPointer** element.

Parent elements
LinearPointers

The following is the XML Schema definition of the **LinearPointers.LinearPointer** element.

```
<xsd:element name="LinearPointer" type="LinearPointerType" minOccurs="1"
  maxOccurs="unbounded" />
```

## 2.182 LinearPointer

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearPointer** element specifies a **linear pointer** to be drawn against a [LinearScale](#) instance. The **LinearPointer** element MUST be specified at least once within a [LinearPointers](#) collection.

The following are the parent elements, attributes, and child elements of the **LinearPointer** element.

Parent elements
<a href="#">LinearPointers</a>

Attributes
<a href="#">LinearPointer.Name</a>

Child elements
<a href="#">LinearPointer.Thermometer</a>
<a href="#">LinearPointer.Type</a>
<a href="#">LinearPointer.ActionInfo</a>
<a href="#">LinearPointer.BarStart</a>
<a href="#">LinearPointer.DistanceFromScale</a>
<a href="#">LinearPointer.GaugeInputValue</a>
<a href="#">LinearPointer.Hidden</a>
<a href="#">LinearPointer.MarkerLength</a>
<a href="#">LinearPointer.MarkerStyle</a>
<a href="#">LinearPointer.Placement</a>
<a href="#">LinearPointer.PointerImage</a>
<a href="#">LinearPointer.SnappingEnabled</a>
<a href="#">LinearPointer.SnappingInterval</a>
<a href="#">LinearPointer.Style</a>
<a href="#">LinearPointer.ToolTip</a>
<a href="#">LinearPointer.Width</a>

The following is the XML Schema definition of the **LinearPointer** element in RDL 2008/01.

```
<xsd:complexType name="LinearPointerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePointerTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

```

<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="GaugeInputValue" type="GaugeInputValueType"
minOccurs="0" />
<xsd:element name="PointerImage" type="PointerImageType" minOccurs="0" />
<xsd:element name="BarStart" type="xsd:string" minOccurs="0" />
<xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
<xsd:element name="MarkerLength" type="xsd:string" minOccurs="0" />
<xsd:element name="MarkerStyle" type="xsd:string" minOccurs="0" />
<xsd:element name="Placement" type="xsd:string" minOccurs="0" />
<xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
<xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
<xsd:element name="Type" type="xsd:string" minOccurs="0" />
<!--GaugePointerTypeEnd-->
  <xsd:element name="Thermometer" type="ThermometerType" minOccurs="0" />
  <xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **LinearPointer** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="LinearPointerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePointerTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType"
minOccurs="0" />
    <xsd:element name="PointerImage" type="PointerImageType" minOccurs="0" />
    <xsd:element name="BarStart" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerLength" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <!--GaugePointerTypeEnd-->
    <xsd:element name="Thermometer" type="ThermometerType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

## 2.182.1 LinearPointer.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearPointer.Name** attribute specifies a unique identifier for a **linear pointer**. This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique within a particular [LinearPointers](#) collection.

The following is the parent element of the **LinearPointer.Name** attribute.

<b>Parent elements</b>
------------------------

<a href="#">LinearPointer</a>
-------------------------------

The following is the XML Schema definition of the **LinearPointer.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

## 2.182.2 LinearPointer.ActionInfo

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearPointer.ActionInfo** element specifies the actions for a [LinearPointer](#). This element is optional. This element is of type [ActionInfo](#).

The following is the parent element of the **LinearPointer.ActionInfo** element.

<b>Parent elements</b>
------------------------

<a href="#">LinearPointer</a>
-------------------------------

The following is the XML Schema definition of the **LinearPointer.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

## 2.182.3 LinearPointer.BarStart

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearPointer.BarStart** element specifies where a [LinearPointer](#) will start if [LinearPointer.Type](#) is set to "Bar". This element is optional.

If the **LinearPointer.BarStart** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/21](#) section 3.2.1) or an expression that evaluates to a **String**. This element is ignored if the associated **LinearPointer.Type** is not set to "Bar". The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**ScaleStart:** The bar starts from the starting point of the [LinearScale](#).

**Zero:** The bar starts from the zero point of the **LinearScale**.

If the **LinearPointer.BarStart** element is not present, its value is interpreted as "ScaleStart".

The following is the parent element of the **LinearPointer.BarStart** element.

<b>Parent elements</b>
------------------------

<a href="#">LinearPointer</a>
-------------------------------

The following is the XML Schema definition of the **LinearPointer.BarStart** element.

```
<xsd:element name="BarStart" type="xsd:string" minOccurs="0" />
```

## 2.182.4 LinearPointer.DistanceFromScale

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearPointer.DistanceFromScale** element specifies the distance from the tip of a [LinearPointer](#) to its associated [LinearScale](#), as a percentage of the size of the **LinearScale**.

The **LinearPointer.DistanceFromScale** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a [Float.<41>](#) If the **LinearPointer.DistanceFromScale** element is not present, its value is interpreted as 0.

The following is the parent element of the **LinearPointer.DistanceFromScale** element.

Parent elements
LinearPointer

The following is the XML Schema definition of the **LinearPointer.DistanceFromScale** element.

```
<xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
```

## 2.182.5 LinearPointer.GaugeInputValue

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearPointer.GaugeInputValue** element specifies the value to use for a [LinearPointer](#). This element is optional. This element is of type [GaugeInputValue](#).

The following is the parent element of the **LinearPointer.GaugeInputValue** element.

Parent elements
LinearPointer

The following is the XML Schema definition of the **LinearPointer.GaugeInputValue** element.

```
<xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="0" />
```

## 2.182.6 LinearPointer.Hidden

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearPointer.Hidden** element specifies whether a [LinearPointer](#) is hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **LinearPointer.Hidden** element.

Parent elements
LinearPointer

The following is the XML Schema definition of the **LinearPointer.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

### 2.182.7 LinearPointer.MarkerLength

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearPointer.MarkerLength** element specifies the length of a marker for a [LinearPointer](#) as a percentage of the length of the parent [LinearScale](#). The **LinearPointer.MarkerLength** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0. [<42>](#) If the **LinearPointer.MarkerLength** element is not present, its value is interpreted as 0.

The following is the parent element of the **LinearPointer.MarkerLength** element.

Parent elements
LinearPointer

The following is the XML Schema definition of the **LinearPointer.MarkerLength** element.

```
<xsd:element name="MarkerLength" type="xsd:string" minOccurs="0" />
```

### 2.182.8 LinearPointer.MarkerStyle

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearPointer.MarkerStyle** element specifies the type of the marker for a [LinearPointer](#). This element is optional. If the **LinearPointer.MarkerStyle** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Triangle:** The marker is of type "Triangle".

**Rectangle:** The marker is of type "Rectangle".

**Circle:** The marker is of type "Circle".

**Diamond:** The marker is of type "Diamond".

**Trapezoid:** The marker is of type "Trapezoid".

**Star:** The marker is of type "Star".

**Wedge:** The marker is of type "Wedge".

**Pentagon:** The marker is of type "Pentagon".

**None:** The marker is of type "None".

If the **LinearPointer.MarkerStyle** element is not present, its value is interpreted as "Triangle".

The following is the parent element of the **LinearPointer.MarkerStyle** element.

Parent elements
LinearPointer

The following is the XML Schema definition of the **LinearPointer.MarkerStyle** element.

```
<xsd:element name="MarkerStyle" type="xsd:string" minOccurs="0" />
```

## 2.182.9 LinearPointer.Placement

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearPointer.Placement** element determines where a [LinearPointer](#) is placed relative to a [LinearScale](#). This element is optional.

If the **LinearPointer.Placement** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Inside:** The pointer is placed inside of the **LinearScale**.

**Outside:** The pointer is placed outside of the **LinearScale**.

**Cross:** The pointer is placed across the **LinearScale**.

If the **LinearPointer.Placement** element is not present, its value is interpreted as "Inside".

The following is the parent element of the **LinearPointer.Placement** element.

Parent elements
LinearPointer

The following is the XML Schema definition of the **LinearPointer.Placement** element.

```
<xsd:element name="Placement" type="xsd:string" minOccurs="0" />
```

## 2.182.10 LinearPointer.PointerImage

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearPointer.PointerImage** element specifies the image to use for a [LinearPointer](#). This element is optional. This element is of type [PointerImage](#).

The following is the parent element of the **LinearPointer.PointerImage** element.

Parent elements
LinearPointer

The following is the XML Schema definition of the **LinearPointer.PointerImage** element.

```
<xsd:element name="PointerImage" type="PointerImageType" minOccurs="0" />
```

## 2.182.11 LinearPointer.SnappingEnabled

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearPointer.SnappingEnabled** element specifies whether values within a [LinearPointer](#) round to the snapping interval. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **LinearPointer.SnappingEnabled** element.

Parent elements
<a href="#">LinearPointer</a>

The following is the XML Schema definition of the **LinearPointer.SnappingEnabled** element.

```
<xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
```

### 2.182.12 LinearPointer.SnappingInterval

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearPointer.SnappingInterval** element specifies the interval to which values within a [LinearPointer](#) round. The **LinearPointer.SnappingInterval** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. The value of the **LinearPointer.SnappingInterval** element MUST be non-negative. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **LinearPointer.SnappingInterval** element.

Parent elements
<a href="#">LinearPointer</a>

The following is the XML Schema definition of the **LinearPointer.SnappingInterval** element.

```
<xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
```

### 2.182.13 LinearPointer.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **LinearPointer.Style** element specifies style properties for a **linear pointer**. This element is optional. This element is of type [Style](#).

The following is the parent element of the **LinearPointer.Style** element.

Parent elements
<a href="#">LinearPointer</a>

The following is the XML Schema definition of the **LinearPointer.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```



## 2.182.14 LinearPointer.ToolTip

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearPointer.ToolTip** element specifies the tooltip text for a [LinearPointer](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **LinearPointer.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **LinearPointer.ToolTip** element.

Parent elements
<a href="#">LinearPointer</a>

The following is the XML Schema definition of the **LinearPointer.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

## 2.182.15 LinearPointer.Width

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearPointer.Width** element specifies the width of a [LinearPointer](#) as a percentage of [LinearScale.Width](#). The **LinearPointer.Width** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **LinearPointer.Width** element.

Parent elements
<a href="#">LinearPointer</a>

The following is the XML Schema definition of the **LinearPointer.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
```

## 2.182.16 LinearPointer.Thermometer

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearPointer.Thermometer** element specifies style properties for a [Thermometer](#) instance. The **LinearPointer.Thermometer** element is optional. This element is of type **Thermometer**.

The following is the parent element of the **LinearPointer.Thermometer** element.

Parent elements
<a href="#">LinearPointer</a>

The following is the XML Schema definition of the **LinearPointer.Thermometer** element.

```
<xsd:element name="Thermometer" type="ThermometerType" minOccurs="0">
```

## 2.182.17 LinearPointer.Type

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **LinearPointer.Type** element specifies the type of a [LinearPointer](#). The **LinearPointer.Type** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Marker**: Specifies that the **linear pointer** is a marker.

**Bar**: Specifies that the linear pointer is a bar.

**Thermometer**: Specifies that the linear pointer is a thermometer.

If the **LinearPointer.Type** element is not present, its value is interpreted as "Marker".

The following is the parent element of the **LinearPointer.Type** element.

Parent elements
<a href="#">LinearPointer</a>

The following is the XML Schema definition of the **LinearPointer.Type** element.

```
<xsd:element name="Type" type="xsd:string" minOccurs="0" />
```

## 2.183 PointerImage

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PointerImage** element specifies a **pointer image** to be used for a [LinearPointer](#) or a [RadialPointer](#) within a [LinearGauge](#) or [RadialGauge](#) instance. The **PointerImage** element is optional.

The following are the parent and child elements of the **PointerImage** element.

Parent elements
<a href="#">LinearPointer</a>
<a href="#">RadialPointer</a>

Child elements
<a href="#">PointerImage.HueColor</a>
<a href="#">PointerImage.OffsetX</a>
<a href="#">PointerImage.OffsetY</a>
<a href="#">PointerImage.Transparency</a>
<a href="#">PointerImage.MIMETYPE</a>
<a href="#">PointerImage.Source</a>
<a href="#">PointerImage.TransparentColor</a>

<b>Child elements</b>
-----------------------

<a href="#">PointerImage.Value</a>
------------------------------------

The following is the XML Schema definition of the **PointerImage** element in RDL 2008/01.

```
<xsd:complexType name="PointerImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="SizeType" minOccurs="0" />
    <xsd:element name="OffsetY" type="SizeType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **PointerImage** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="PointerImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="SizeType" minOccurs="0" />
    <xsd:element name="OffsetY" type="SizeType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.183.1 PointerImage.MIMETYPE

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PointerImage.MIMETYPE** element specifies the image format of a [PointerImage](#). The **PointerImage.MIMETYPE** element is optional. If this element is present, its value **MUST** be a [ReportMIMETYPE](#).

If the peer element [PointerImage.Source](#) is set to a value other than "Database", the **PointerImage.MIMETYPE** element is ignored.

The following is the parent element of the **PointerImage.MIMETYPE** element.

<b>Parent elements</b>
------------------------

<a href="#">PointerImage</a>
------------------------------

The following is the XML Schema definition of the **PointerImage.MIMETYPE** element.

```
<xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0">
```

### 2.183.2 PointerImage.Source

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PointerImage.Source** element specifies the type of source associated with a [PointerImage](#). The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**External:** Specifies that the peer [PointerImage.Value](#) element contains a **String** constant or expression that evaluates to the location of an image.

**Embedded:** Specifies that the peer **PointerImage.Value** element contains a **String** constant or expression that evaluates to the name of an [EmbeddedImage](#) within the report.

**Database:** Specifies that the peer **PointerImage.Value** element contains an expression (for example, a field in the database) that evaluates to the binary data for an image.

The **PointerImage.Source** element MUST be specified.

Parent elements
PointerImage

The following is the XML Schema definition of the **PointerImage.Source** element.

```
<xsd:element name="Source" type="xsd:string" minOccurs="1">
```

### 2.183.3 PointerImage.TransparentColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PointerImage.TransparentColor** element specifies the color to treat as transparent in a [PointerImage](#). This element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **PointerImage.TransparentColor** element.

Parent elements
PointerImage

The following is the XML Schema definition of the **PointerImage.TransparentColor** element.

```
<xsd:element name="TransparentColor" type="xsd:string" minOccurs="0">
```

### 2.183.4 PointerImage.Value

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PointerImage.Value** element specifies the location of a [PointerImage](#) depending on the peer [PointerImage.Source](#) element. The **PointerImage.Value** element MUST be specified.

If the peer **PointerImage.Source** element is set to "External" and if the value of **PointerImage.Value** is non-empty, then the value of **PointerImage.Value** MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) constant or an expression that evaluates to the location of an image. This location MUST be a [ReportPath](#) or [RdlColor](#).

If the peer **PointerImage.Source** element is set to "Embedded" and if the value of **PointerImage.Value** is non-empty, then the value of **PointerImage.Value** MUST be a **String** constant or an expression that evaluates to the name of an [EmbeddedImage](#) within the report.

If the peer **PointerImage.Source** element is set to "Database" and if its value is non-empty, its value MUST be an expression that evaluates to the binary data for an image. If the **PointerImage.Value** element has an empty value, the image MUST NOT be displayed.

The following is the parent element of the **PointerImage.Value** element.

Parent elements
PointerImage

The following is the XML Schema definition of the **PointerImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1">
```

### 2.183.5 PointerImage.HueColor

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PointerImage.HueColor** element specifies the color to tint a [PointerImage](#). This element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **PointerImage.HueColor** element.

Parent elements
PointerImage

The following is the XML Schema definition of the **PointerImage.HueColor** element.

```
<xsd:element name="HueColor" type="xsd:string" minOccurs="0">
```

### 2.183.6 PointerImage.OffsetX

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PointerImage.OffsetX** element specifies the X, or horizontal, offset for a [PointerImage](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#).

The following is the parent element of the **PointerImage.OffsetX** element.

Parent elements
PointerImage

The following is the XML Schema definition of the **PointerImage.OffsetX** element.

```
<xsd:element name="OffsetX" type="SizeType" minOccurs="0">
```

### 2.183.7 PointerImage.OffsetY

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PointerImage.OffsetY** element specifies the Y, or vertical, offset for a [PointerImage](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#).

The following is the parent element of the **PointerImage.OffsetY** element.

Parent elements
PointerImage

The following is the XML Schema definition of the **PointerImage.OffsetY** element.

```
<xsd:element name="OffsetY" type="SizeType" minOccurs="0">
```

### 2.183.8 PointerImage.Transparency

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PointerImage.Transparency** element specifies the percentage of transparency for a [PointerImage](#). This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0 and less than or equal to 100. If the **PointerImage.Transparency** element is not present, its value is interpreted as "0".

The following is the parent element of the **PointerImage.Transparency** element.

Parent elements
PointerImage

The following is the XML Schema definition of the **PointerImage.Transparency** element.

```
<xsd:element name="Transparency" type="xsd:string" minOccurs="0">
```

### 2.184 Thermometer

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Thermometer** element specifies display properties for a [LinearPointer.Thermometer](#) instance that has the value of the [LinearPointer.Type](#) element set to "Thermometer". The **Thermometer** element is optional. This element is ignored if the parent [LinearPointer](#) element does not have the value of the **LinearPointer.Type** element set to "Thermometer".

The following are the parent and child elements of the **Thermometer** element.

Parent elements
LinearPointer

Child elements
<a href="#">Thermometer.BulbOffset</a>
<a href="#">Thermometer.BulbSize</a>
<a href="#">Thermometer.Style</a>
<a href="#">Thermometer.ThermometerStyle</a>

The following is the XML Schema definition of the **Thermometer** element in RDL 2008/01.

```
<xsd:complexType name="ThermometerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="BulbOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="BulbSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="ThermometerStyle" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **Thermometer** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ThermometerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="BulbOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="BulbSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="ThermometerStyle" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.184.1 Thermometer.BulbOffset

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Thermometer.BulbOffset** element specifies the offset of the bulb of a thermometer [LinearPointer](#) instance as a percentage of the length of a parent [LinearScale](#) instance.

The **Thermometer.BulbOffset** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0. [<43>](#) If this element is not present, its value is interpreted as 5.

The following is the parent element of the **Thermometer.BulbOffset** element.

<b>Parent elements</b>
------------------------

<a href="#">Thermometer</a>
-----------------------------

The following is the XML Schema definition of the **Thermometer.BulbOffset** element.

```
<xsd:element name="BulbOffset" type="xsd:string" minOccurs="0">
```

## 2.184.2 Thermometer.BulbSize

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Thermometer.BulbSize** element specifies the size of the bulb of a thermometer [LinearPointer](#) instance as a percentage of the length of a parent [LinearScale](#) instance. The **Thermometer.BulbSize** element is optional.

If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0. [<44>](#) If this element is not present, its value is interpreted as 50.

The following is the parent element of the **Thermometer.BulbSize** element.

<b>Parent elements</b>
------------------------

<a href="#">Thermometer</a>
-----------------------------

The following is the XML Schema definition of the **Thermometer.BulbSize** element.

```
<xsd:element name="BulbSize" type="xsd:string" minOccurs="0">
```

## 2.184.3 Thermometer.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Thermometer.Style** element specifies style properties for a thermometer [LinearPointer](#) instance. This element is optional. This element is of type [Style](#).

The following is the parent element of the **Thermometer.Style** element.

<b>Parent elements</b>
------------------------

<a href="#">Thermometer</a>
-----------------------------

The following is the XML Schema definition of the **Thermometer.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0">
```

## 2.184.4 Thermometer.ThermometerStyle

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Thermometer.ThermometerStyle** element specifies the type of the thermometer that is used for a thermometer. This element is optional.



If the **Thermometer.ThermometerStyle** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Standard:** Specifies that the thermometer has a standard look with a circular bulb.

**Flask:** Specifies that the thermometer looks like a flask.

If the **Thermometer.ThermometerStyle** element is not present, its value is interpreted as "Standard".

The following is the parent element of the **Thermometer.ThermometerStyle** element.

Parent elements
<a href="#">Thermometer</a>

The following is the XML Schema definition of the **Thermometer.ThermometerStyle** element.

```
<xsd:element name="ThermometerStyle" type="xsd:string" minOccurs="0">
```

## 2.185 ScaleLabels

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleLabels** element specifies the appearance of labels on a [RadialScale](#) or [LinearScale](#). This element is optional.

The following are the parent and child elements of the **ScaleLabels** element.

Parent elements
<a href="#">LinearScale</a>
<a href="#">RadialScale</a>

Child elements
<a href="#">ScaleLabels.AllowUpsideDown</a>
<a href="#">ScaleLabels.DistanceFromScale</a>
<a href="#">ScaleLabels.FontAngle</a>
<a href="#">ScaleLabels.Hidden</a>
<a href="#">ScaleLabels.Interval</a>
<a href="#">ScaleLabels.IntervalOffset</a>
<a href="#">ScaleLabels.Placement</a>
<a href="#">ScaleLabels.RotateLabels</a>
<a href="#">ScaleLabels.ShowEndLabels</a>
<a href="#">ScaleLabels.Style</a>

<b>Child elements</b>
-----------------------

<a href="#">ScaleLabels.UseFontPercent</a>
--

The following is the XML Schema definition of the **ScaleLabels** element in RDL 2008/01.

```
<xsd:complexType name="ScaleLabelsType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="RotateLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowEndLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ScaleLabels** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ScaleLabelsType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="RotateLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowEndLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.185.1 ScaleLabels.AllowUpsideDown

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleLabels.AllowUpsideDown** element specifies whether the labels within a [RadialScale](#) or [LinearScale](#) can be rotated by a value greater than 90 degrees.

This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **ScaleLabels.AllowUpsideDown** element.

<b>Parent elements</b>
------------------------

<a href="#">ScaleLabels</a>
-----------------------------

The following is the XML Schema definition of the **ScaleLabels.AllowUpsideDown** element.

```
<xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0">
```

### 2.185.2 ScaleLabels.DistanceFromScale

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScaleLabels.DistanceFromScale** element specifies the distance from the labels to a [RadialScale](#) or [LinearScale](#) as a percentage of the size, specified by the radius in a **radial scale** or the length in a **linear scale**.

This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**.[<45>](#) If this element is not present, its value is interpreted as "2".

The following is the parent element of the **ScaleLabels.DistanceFromScale** element.

Parent elements
<a href="#">ScaleLabels</a>

The following is the XML Schema definition of the **ScaleLabels.DistanceFromScale** element.

```
<xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0">
```

### 2.185.3 ScaleLabels.FontAngle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScaleLabels.FontAngle** element specifies, in degrees, the angle of rotation for text in labels within a [RadialScale](#) or [LinearScale](#). This element is optional. If the **ScaleLabels.FontAngle** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0 and less than or equal to 360. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **ScaleLabels.FontAngle** element.

Parent elements
<a href="#">ScaleLabels</a>

The following is the XML Schema definition of the **ScaleLabels.FontAngle** element.

```
<xsd:element name="FontAngle" type="xsd:string" minOccurs="0">
```

### 2.185.4 ScaleLabels.Hidden

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScaleLabels.Hidden** element specifies whether the labels within a [RadialScale](#) or [LinearScale](#) are hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **ScaleLabels.Hidden** element.

Parent elements
<a href="#">ScaleLabels</a>

The following is the XML Schema definition of the **ScaleLabels.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0">
```

### 2.185.5 ScaleLabels.Interval

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleLabels.Interval** element specifies the intervals between labels in a [RadialScale](#) or [LinearScale](#). This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be non-negative. If the **ScaleLabels.Interval** element is not present, its value is interpreted as 0, which is interpreted to be the same value as that of the [RadialScale.Interval](#) or [LinearScale.Interval](#) element that is associated with this [ScaleLabels](#) instance.

The following is the parent element of the **ScaleLabels.Interval** element.

Parent elements
<a href="#">ScaleLabels</a>

The following is the XML Schema definition of the **ScaleLabels.Interval** element.

```
<xsd:element name="Interval" type="xsd:string" minOccurs="0">
```

### 2.185.6 ScaleLabels.IntervalOffset

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleLabels.IntervalOffset** element specifies the interval offset for the first label within a [RadialScale](#) or [LinearScale](#). This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be non-negative. If this element is not present, its value is interpreted as 0, which is interpreted to be the same value as that of the [RadialScale.IntervalOffset](#) or [LinearScale.Interval](#) element that is associated with this [ScaleLabels](#) instance.

The following is the parent element of the **ScaleLabels.IntervalOffset** element.

Parent elements
<a href="#">ScaleLabels</a>

The following is the XML Schema definition of the **ScaleLabels.IntervalOffset** element.

```
<xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0">
```

## 2.185.7 ScaleLabels.Placement

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleLabels.Placement** element specifies where the labels in a [RadialScale](#) or [LinearScale](#) are placed. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Inside:** The labels are placed inside the [GaugeTickMarks](#) instances within the **radial scale** or **linear scale**.

**Outside:** The labels are placed outside the **GaugeTickMarks** instances within the radial scale or linear scale.

**Cross:** The labels are placed across the **GaugeTickMarks** instances within the radial scale or linear scale.

If the **ScaleLabels.Placement** element is not present, its value is interpreted as "Inside".

The following is the parent element of the **ScaleLabels.Placement** element.

Parent elements
<a href="#">ScaleLabels</a>

The following is the XML Schema definition of the **ScaleLabels.Placement** element.

```
<xsd:element name="Placement" type="xsd:string" minOccurs="0">
```

## 2.185.8 ScaleLabels.RotateLabels

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleLabels.RotateLabels** element indicates whether the text for [ScaleLabels](#) rotates along with the associated [RadialScale](#) or [LinearScale](#). The **ScaleLabels.RotateLabels** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **ScaleLabels.RotateLabels** element.

Parent elements
ScaleLabels

The following is the XML Schema definition of the **ScaleLabels.RotateLabels** element.

```
<xsd:element name="RotateLabels" type="xsd:string" minOccurs="0">
```

## 2.185.9 ScaleLabels.ShowEndLabels

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleLabels.ShowEndLabels** element indicates whether the labels at the end of a [RadialScale](#) or [LinearScale](#) are shown. The **ScaleLabels.ShowEndLabels** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its **Boolean** value is interpreted as false.

The following is the parent element of the **ScaleLabels.ShowEndLabels** element.

Parent elements
<a href="#">ScaleLabels</a>

The following is the XML Schema definition of the **ScaleLabels.ShowEndLabels** element.

```
<xsd:element name="ShowEndLabels" type="xsd:string" minOccurs="0">
```

### 2.185.10 ScaleLabels.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScaleLabels.Style** element specifies style properties for labels within a [RadialScale](#) or [LinearScale](#). This element is optional. This element is of type [Style](#).

The following is the parent element of the **ScaleLabels.Style** element.

Parent elements
<a href="#">ScaleLabels</a>

The following is the XML Schema definition of the **ScaleLabels.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0">
```

### 2.185.11 ScaleLabels.UseFontPercent

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScaleLabels.UseFontPercent** element specifies whether the font sizes for the labels within a [RadialScale](#) or a [LinearScale](#) are measured as a percentage of the parent font sizes or in the units specified by the [Style.FontSize](#) element of the [ScaleLabels.Style](#) element. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **ScaleLabels.UseFontPercent** element.

Parent elements
<a href="#">ScaleLabels</a>

The following is the XML Schema definition of the **ScaleLabels.UseFontPercent** element.

```
<xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0">
```

## 2.186 ScalePin

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScalePin** element specifies a scale pin at one end of a [LinearScale](#) or [RadialScale](#) instance. The **ScalePin** element is optional. This element MUST be represented by one of the following:

- [LinearScale.MaximumPin](#)
- [LinearScale.MinimumPin](#)
- [RadialScale.MaximumPin](#)
- [RadialScale.MinimumPin](#)

The following are the parent and child elements of the **ScalePin** element.

Parent elements
<a href="#">LinearScale</a>
<a href="#">RadialScale</a>

Child elements
<a href="#">ScalePin.DistanceFromScale</a>
<a href="#">ScalePin.EnableGradient</a>
<a href="#">ScalePin.GradientDensity</a>
<a href="#">ScalePin.Hidden</a>
<a href="#">ScalePin.Length</a>
<a href="#">ScalePin.Placement</a>
<a href="#">ScalePin.Shape</a>
<a href="#">ScalePin.Style</a>
<a href="#">ScalePin.TickMarkImage</a>
<a href="#">ScalePin.Width</a>
<a href="#">ScalePin.Enable</a>
<a href="#">ScalePin.Location</a>
<a href="#">ScalePin.PinLabel</a>

The following is the XML Schema definition of the **ScalePin** element in RDL 2008/01.

```
<xsd:complexType name="ScalePinType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--TickMarkStyleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
  </choice>
</complexType>
```

```

<xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
<xsd:element name="Length" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
<xsd:element name="Shape" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<!--TickMarkStyleTypeEnd-->
  <xsd:element name="Location" type="xsd:string" minOccurs="0" />
  <xsd:element name="Enable" type="xsd:string" minOccurs="0" />
  <xsd:element name="PinLabel" type="PinLabelType" minOccurs="0" />
  <xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **ScalePin** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="ScalePinType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--TickMarkStyleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
    <xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shape" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <!--TickMarkStyleTypeEnd-->
    <xsd:element name="Location" type="xsd:string" minOccurs="0" />
    <xsd:element name="Enable" type="xsd:string" minOccurs="0" />
    <xsd:element name="PinLabel" type="PinLabelType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

## 2.186.1 ScalePin.DistanceFromScale

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScalePin.DistanceFromScale** element specifies the distance from the [ScalePin](#) to a containing [RadialScale](#) or [LinearScale](#). The **ScalePin.DistanceFromScale** element is optional.

If this element is present, its value MUST be a [Float \[IEEE754\]](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**.[<46>](#) If this element is not present, its value is interpreted as 0.

The following is the parent element of the **ScalePin.DistanceFromScale** element.

Parent elements
ScalePin

The following is the XML Schema definition of the **ScalePin.DistanceFromScale** element.

```

<xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0">

```



## 2.186.2 ScalePin.EnableGradient

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScalePin.EnableGradient** element specifies whether a gradient effect is used for a [ScalePin](#). The **ScalePin.EnableGradient** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ScalePin.EnableGradient** element.

Parent elements
ScalePin

The following is the XML Schema definition of the **ScalePin.EnableGradient** element.

```
<xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
```

## 2.186.3 ScalePin.GradientDensity

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScalePin.GradientDensity** element specifies the intensity of the gradient effect for a [ScalePin](#). This element is optional.

If the **ScalePin.GradientDensity** element is present, its value MUST be a [Float \[IEEE754\]](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0 and less than or equal to 100, specifying the percentage of maximum density. If this element is not present, its value is interpreted as 30.

Following is the parent element of the **ScalePin.GradientDensity** element.

Parent elements
ScalePin

The following is the XML Schema definition of the **ScalePin.GradientDensity** element.

```
<xsd:element name="GradientDensity" type="xsd:string" minOccurs="0">
```

## 2.186.4 ScalePin.Hidden

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScalePin.Hidden** element specifies whether a [ScalePin](#) is hidden. The **ScalePin.Hidden** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ScalePin.Hidden** element.

Parent elements
ScalePin

The following is the XML Schema definition of the **ScalePin.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0">
```

## 2.186.5 ScalePin.Length

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScalePin.Length** element specifies the length of the [ScalePin](#) as a percentage of the radius of a containing [RadialScale](#) or the minimum of the length and width of a containing [LinearScale](#). This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0. [<47>](#) If this element is not present, its value is interpreted as 0.

The following is the parent element of the **ScalePin.Length** element.

Parent elements
ScalePin

The following is the XML Schema definition of the **ScalePin.Length** element.

```
<xsd:element name="Length" type="xsd:string" minOccurs="0">
```

## 2.186.6 ScalePin.Placement

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScalePin.Placement** element specifies where the [ScalePin](#) is placed relative to its associated [RadialScale](#) or [LinearScale](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Inside:** The **ScalePin** is placed inside the [GaugeTickMarks](#) instances in the **radial scale** or **linear scale**.

**Outside:** The **ScalePin** is placed outside the **GaugeTickMarks** instances within the radial scale or linear scale.

**Cross:** The **ScalePin** is placed across the **GaugeTickMarks** instances within the radial scale or linear scale.

If the **ScalePin.Placement** element is not present, its value is interpreted as "Inside".

The following is the parent element of the **ScalePin.Placement** element.

Parent elements
ScalePin

The following is the XML Schema definition of the **ScalePin.Placement** element.

```
<xsd:element name="Placement" type="xsd:string" minOccurs="0">
```

## 2.186.7 ScalePin.Shape

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScalePin.Shape** element specifies the shape of a [ScalePin](#). This element is optional. If the **ScalePin.Shape** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Rectangle:** The marker is of type "Rectangle".

**Triangle:** The marker is of type "Triangle".

**Circle:** The marker is of type "Circle".

**Diamond:** The marker is of type "Diamond".

**Trapezoid:** The marker is of type "Trapezoid".

**Star:** The marker is of type "Star".

**Wedge:** The marker is of type "Wedge".

**Pentagon:** The marker is of type "Pentagon".

**None:** The marker is of type "None".

If the **ScalePin.Shape** element is not present, its value is interpreted as "Rectangle".

The following is the parent element of the **ScalePin.Shape** element.

Parent elements
ScalePin

The following is the XML Schema definition of the **ScalePin.Shape** element.

```
<xsd:element name="Shape" type="xsd:string" minOccurs="0">
```

## 2.186.8 ScalePin.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScalePin.Style** element specifies style properties for the [ScalePin](#). The **ScalePin.Style** element is optional. This element is of type [Style](#).

The following is the parent element of the **ScalePin.Style** element.

Parent elements
ScalePin

The following is the XML Schema definition of the **ScalePin.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0">
```

## 2.186.9 ScalePin.TickMarkImage

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScalePin.TickMarkImage** element specifies the image to use for the [ScalePin](#). The **ScalePin.TickMarkImage** element is optional. This element is of type [TopImage](#).

The following is the parent element of the **ScalePin.TickMarkImage** element.

Parent elements
ScalePin

The following is the XML Schema definition of the **ScalePin.TickMarkImage** element.

```
<xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0">
```

## 2.186.10 ScalePin.Width

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScalePin.Width** element specifies the width of the [ScalePin](#) as a percentage of the radius of a containing [RadialScale](#) or the minimum of the length and width of a containing [LinearScale](#). This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0. [<48>](#) If this element is not present, its value is interpreted as 0.

Following is the parent element of the **ScalePin.Width** element.

Parent elements
ScalePin

The following is the XML Schema definition of the **ScalePin.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0">
```

## 2.186.11 ScalePin.Enable

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScalePin.Enable** element specifies that a [ScalePin](#) instance is enabled. The **ScalePin.Enable** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ScalePin.Enable** element.

Parent elements
ScalePin

The following is the XML Schema definition of the **ScalePin.Enable** element.

```
<xsd:element name="Enable" type="xsd:string" minOccurs="0">
```

### 2.186.12 ScalePin.Location

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScalePin.Location** element specifies the location of a [ScalePin](#), relative to the start or end of the containing [RadialScale](#) or [LinearScale](#).

The **ScalePin.Location** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element is interpreted in units of degrees for a **radial scale** and in terms of percentage for a **linear scale**. If this element is not present, its value is interpreted as "5".

The following is the parent element of the **ScalePin.Location** element.

Parent elements
ScalePin

The following is the XML Schema definition of the **ScalePin.Location** element.

```
<xsd:element name="Location" type="xsd:string" minOccurs="0">
```

### 2.186.13 ScalePin.PinLabel

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScalePin.PinLabel** element specifies a label for a [ScalePin](#). The **ScalePin.PinLabel** element is optional. This element is of type [PinLabel](#).

The following is the parent element of the **ScalePin.PinLabel** element.

Parent elements
ScalePin

The following is the XML Schema definition of the **ScalePin.PinLabel** element.

```
<xsd:element name="PinLabel" type="PinLabelType" minOccurs="0">
```

### 2.187 PinLabel

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PinLabel** element specifies a label for a [ScalePin](#). This element is optional.

The following are the parent and child elements of the **PinLabel** element.

Parent elements
ScalePin

Child elements
<a href="#">PinLabel.AllowUpsideDown</a>
<a href="#">PinLabel.DistanceFromScale</a>
<a href="#">PinLabel.FontAngle</a>
<a href="#">PinLabel.Placement</a>
<a href="#">PinLabel.RotateLabel</a>
<a href="#">PinLabel.Style</a>
<a href="#">PinLabel.Text</a>
<a href="#">PinLabel.UseFontPercent</a>

The following is the XML Schema definition of the **PinLabel** element in RDL 2008/01.

```
<xsd:complexType name="PinLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="RotateLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **PinLabel** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="PinLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="RotateLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.187.1 PinLabel.AllowUpsideDown

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PinLabel.AllowUpsideDown** element specifies whether a [PinLabel](#) instance can be rotated by more than 90 degrees. This element is optional. This element is ignored if the associated **PinLabel** instance is not within a [RadialScale](#).

If the **PinLabel.AllowUpsideDown** element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **PinLabel.AllowUpsideDown** element.

Parent elements
PinLabel

The following is the XML Schema definition of the **PinLabel.AllowUpsideDown** element.

```
<xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0">
```

## 2.187.2 PinLabel.DistanceFromScale

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PinLabel.DistanceFromScale** element specifies the distance of a [PinLabel](#) instance to an associated [RadialScale](#) or [LinearScale](#) element, as a percentage of the radius for **RadialScale** instances or length for **LinearScale** instances.

The **PinLabel.DistanceFromScale** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**.[<49>](#) If this element is not present, its value is interpreted as 0.

The following is the parent element of the **PinLabel.DistanceFromScale** element.

Parent elements
PinLabel

The following is the XML Schema definition of the **PinLabel.DistanceFromScale** element.

```
<xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0">
```

## 2.187.3 PinLabel.FontAngle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PinLabel.FontAngle** element specifies, in degrees, the angle of rotation for the text within a [PinLabel](#) instance. The **PinLabel.FontAngle** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0 and less than or equal to 360. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **PinLabel.FontAngle** element.

Parent elements
PinLabel

The following is the XML Schema definition of the **PinLabel.FontAngle** element.

```
<xsd:element name="FontAngle" type="xsd:string" minOccurs="0">
```

## 2.187.4 PinLabel.Placement

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PinLabel.Placement** element specifies where a [PinLabel](#) instance is placed relative to a [RadialScale](#) or [LinearScale](#) element. The **PinLabel.Placement** element is optional.

If the **PinLabel.Placement** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Inside:** The **PinLabel** instance is placed inside the [GaugeTickMarks](#) instances within the **RadialScale** or **LinearScale** instance.

**Outside:** The **PinLabel** instance is placed outside the **GaugeTickMarks** instances within the **RadialScale** or **LinearScale** instance.

**Cross:** The **PinLabel** instance is across the **GaugeTickMarks** instances within the **RadialScale** or **LinearScale** instance.

If the **PinLabel.Placement** element is not present, its value is interpreted as "Inside".

The following is the parent element of the **PinLabel.Placement** element.

Parent elements
PinLabel

The following is the XML Schema definition of the **PinLabel.Placement** element.

```
<xsd:element name="Placement" type="xsd:string" minOccurs="0">
```

## 2.187.5 PinLabel.RotateLabel

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PinLabel.RotateLabel** element specifies whether a [PinLabel](#) instance rotates along with its associated [RadialScale](#) or [LinearScale](#) element. The **PinLabel.RotateLabel** element is ignored if the associated **PinLabel** element is not a descendant of a **RadialScale** element.

The **PinLabel.RotateLabel** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **PinLabel.RotateLabel** element.



Parent elements
PinLabel

The following is the XML Schema definition of the **PinLabel.RotateLabel** element.

```
<xsd:element name="RotateLabel" type="xsd:string" minOccurs="0">
```

## 2.187.6 PinLabel.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PinLabel.Style** element specifies style properties for a [PinLabel](#) instance. The **PinLabel.Style** element is of type [Style](#).

The following is the parent element of the **PinLabel.Style** element.

Parent elements
PinLabel

The following is the XML Schema definition of the **PinLabel.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0">
```

## 2.187.7 PinLabel.Text

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PinLabel.Text** element specifies the text for a [PinLabel](#) instance. The **PinLabel.Text** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **PinLabel.Text** element.

Parent elements
PinLabel

The following is the XML Schema definition of the **PinLabel.Text** element.

```
<xsd:element name="Text" type="xsd:string" minOccurs="0">
```

## 2.187.8 PinLabel.UseFontPercent

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PinLabel.UseFontPercent** element specifies whether the font size for a [PinLabel](#) instance is measured as a percentage of the font size of its ancestor [RadialScale](#) or [LinearScale](#) element or, if set to false, is measured in the units specified by the [Style.FontSize](#) element of the [PinLabel.Style](#) element.

The **PinLabel.UseFontPercent** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **PinLabel.UseFontPercent** element.

Parent elements
PinLabel

The following is the XML Schema definition of the **PinLabel.UseFontPercent** element.

```
<xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0">
```

## 2.188 ScaleRanges

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScaleRanges** element specifies a set of [ScaleRange](#) instances for a [RadialScale](#) or a [LinearScale](#). The **ScaleRanges** element MUST contain at least one [ScaleRanges.ScaleRange](#) instance.

The following are the parent and child elements of the **ScaleRanges** element.

Parent elements
LinearScale
RadialScale

Child elements
ScaleRanges.ScaleRange

The following is the XML Schema definition of the **ScaleRanges** element in RDL 2008/01.

```
<xsd:complexType name="ScaleRangesType">
  <xsd:sequence>
    <xsd:element name="ScaleRange" type="ScaleRangeType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ScaleRanges** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ScaleRangesType">
  <xsd:sequence>
    <xsd:element name="ScaleRange" type="ScaleRangeType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.188.1 ScaleRanges.ScaleRange

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleRanges.ScaleRange** element specifies a range to display on a [RadialScale](#) or a [LinearScale](#). This element MUST be specified. This element is of type [ScaleRange](#).

The following is the parent element of the **ScaleRanges.ScaleRange** element.

Parent elements
<a href="#">ScaleRanges</a>

The following is the XML Schema definition of the **ScaleRanges.ScaleRange** element.

```
<xsd:element name="ScaleRange" type="ScaleRangeType" minOccurs="1"
maxOccurs="unbounded" />
```

## 2.189 ScaleRange

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleRange** element specifies a range to be drawn against a [RadialScale](#) or a [LinearScale](#). This element MUST be specified at least once within a [ScaleRanges](#) collection.

The following are the parent elements, attributes, and child elements of the **ScaleRange** element.

Parent elements
<a href="#">ScaleRanges</a>

Attributes
<a href="#">ScaleRange.Name</a>

Child elements
<a href="#">ScaleRange.ActionInfo</a>
<a href="#">ScaleRange.BackgroundGradientType</a>
<a href="#">ScaleRange.DistanceFromScale</a>
<a href="#">ScaleRange.EndValue</a>
<a href="#">ScaleRange.EndWidth</a>
<a href="#">ScaleRange.Hidden</a>
<a href="#">ScaleRange.InRangeBarPointerColor</a>
<a href="#">ScaleRange.InRangeLabelColor</a>
<a href="#">ScaleRange.InRangeTickMarksColor</a>

Child elements
<a href="#">ScaleRange.Placement</a>
<a href="#">ScaleRange.StartValue</a>
<a href="#">ScaleRange.StartWidth</a>
<a href="#">ScaleRange.Style</a>
<a href="#">ScaleRange.ToolTip</a>

The following is the XML Schema definition of the **ScaleRange** element in RDL 2008/01.

```
<xsd:complexType name="ScaleRangeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="BackgroundGradientType" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="StartWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="InRangeBarPointerColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="InRangeLabelColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="InRangeTickMarksColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ScaleRange** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ScaleRangeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="BackgroundGradientType" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="StartWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="InRangeBarPointerColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="InRangeLabelColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="InRangeTickMarksColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.189.1 ScaleRange.Name

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleRange.Name** attribute specifies the name for a [ScaleRange](#). This attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique within a [ScaleRanges](#) collection.

The following is the parent element of the **ScaleRange.Name** attribute.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

## 2.189.2 ScaleRange.ActionInfo

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleRange.ActionInfo** element specifies the action properties for a [ScaleRange](#) instance. The **ScaleRange.ActionInfo** element is optional. This element is of type [ActionInfo](#).

The following is the parent element of the **ScaleRange.ActionInfo** element.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0">
```

## 2.189.3 ScaleRange.BackgroundGradientType

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleRange.BackgroundGradientType** element specifies the type of background gradient to use against a [ScaleRange](#) instance. The **ScaleRange.BackgroundGradientType** element is optional.

If the **ScaleRange.BackgroundGradientType** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. This element MUST be used instead of the [Style.BackgroundGradientType](#) element for a **ScaleRange** instance. The value of the **ScaleRange.BackgroundGradientType** element MUST be one of the following:

**StartToEnd:** Specifies a gradient from the start of the range to the end of the range.

**LeftRight:** Specifies a gradient from left to right.

**TopBottom:** Specifies a gradient from top to bottom.

**Center:** Specifies a gradient spread out from the center.

**DiagonalLeft:** Specifies a gradient from top left to bottom right.

**DiagonalRight:** Specifies a gradient from top right to bottom left.

**HorizontalCenter:** Specifies a gradient from center to left and right.

**VerticalCenter:** Specifies a gradient from center to top and bottom.

**None:** Specifies that the gradient is disabled.

If the **ScaleRange.BackgroundGradientType** element is not present, its value is interpreted as "StartToEnd".

The following is the parent element of the **ScaleRange.BackgroundGradientType** element.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.BackgroundGradientType** element.

```
<xsd:element name="BackgroundGradientType" type="xsd:string" minOccurs="0" />
```

## 2.189.4 ScaleRange.DistanceFromScale

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleRange.DistanceFromScale** element specifies the distance from the [ScaleRange](#) instance to the containing [RadialScale](#) or [LinearScale](#), as a percentage of the size of the radial scale or linear scale.

The **ScaleRange.DistanceFromScale** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. [<50>](#) If this element is not present, its value is interpreted as 10.

The following is the parent element of the **ScaleRange.DistanceFromScale** element.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.DistanceFromScale** element.

```
<xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0">
```

## 2.189.5 ScaleRange.EndValue

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleRange.EndValue** element specifies the ending value of a [ScaleRange](#) instance. The **ScaleRange.EndValue** element is optional. This element is of type [GaugeInputValue](#).

The following is the parent element of the **ScaleRange.EndValue** element.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.EndValue** element.

```
<xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0">
```

## 2.189.6 ScaleRange.EndWidth

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScaleRange.EndWidth** element specifies the width of a [ScaleRange](#) at its end, as a percentage of the size of the parent [RadialScale](#) or [LinearScale](#) element. The **ScaleRange.EndWidth** element is optional.

If the **ScaleRange.EndWidth** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0. [<51>](#) If this element is not present, its value is interpreted as 0.

The following is the parent element of the **ScaleRange.EndWidth** element.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.EndWidth** element.

```
<xsd:element name="EndWidth" type="xsd:string" minOccurs="0">
```

## 2.189.7 ScaleRange.Hidden

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScaleRange.Hidden** element specifies whether a [ScaleRange](#) instance is hidden. The **ScaleRange.Hidden** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **ScaleRange.Hidden** element.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0">
```

## 2.189.8 ScaleRange.InRangeBarPointerColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScaleRange.InRangeBarPointerColor** element specifies the color of a "bar" type [LinearPointer](#) or [RadialPointer](#) if the linear pointer or **radial pointer** falls within a [ScaleRange](#) instance within the same [RadialScale](#) or [LinearScale](#) element.

The **ScaleRange.InRangeBarPointerColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **ScaleRange.InRangeBarPointerColor** element.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.InRangeBarPointerColor** element.

```
<xsd:element name="InRangeBarPointerColor" type="xsd:string" minOccurs="0" />
```

## 2.189.9 ScaleRange.InRangeLabelColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScaleRange.InRangeLabelColor** element specifies the color of any labels that are specified by a [ScaleLabels](#) instance that falls within a [ScaleRange](#) instance. The **ScaleRange.InRangeLabelColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **ScaleRange.InRangeLabelColor** element.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.InRangeLabelColor** element.

```
<xsd:element name="InRangeLabelColor" type="xsd:string" minOccurs="0">
```

## 2.189.10 ScaleRange.InRangeTickMarksColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScaleRange.InRangeTickMarksColor** element specifies the color of tick marks that fall within a [ScaleRange](#). The **ScaleRange.InRangeTickMarksColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **ScaleRange.InRangeTickMarksColor** element.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.InRangeTickMarksColor** element.

```
<xsd:element name="InRangeTickMarksColor" type="xsd:string" minOccurs="0" />
```

## 2.189.11 ScaleRange.Placement

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*



The **ScaleRange.Placement** element specifies where the [ScaleRange](#) in a [RadialScale](#) or [LinearScale](#) is placed. This element is optional.

If the **ScaleRange.Placement** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/21](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Inside:** The **ScaleRange** instance is placed inside [GaugeTickMarks](#) instances within the **RadialScale** or **LinearScale** element.

**Outside:** The **ScaleRange** instance is placed outside **GaugeTickMarks** instances within the **RadialScale** or **LinearScale** element.

**Cross:** The **ScaleRange** instance is across **GaugeTickMarks** instances within the **RadialScale** or **LinearScale** element.

If the **ScaleRange.Placement** element is not present, its value is interpreted as "Inside".

The following is the parent element of the **ScaleRange.Placement** element.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.Placement** element.

```
<xsd:element name="Placement" type="xsd:string" minOccurs="0">
```

### 2.189.12 ScaleRange.StartValue

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScaleRange.StartValue** element specifies the starting value for a [ScaleRange](#). The **ScaleRange.StartValue** element is optional. This element is of type [GaugeInputValue](#).

The following is the parent element of the **ScaleRange.StartValue** element.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.StartValue** element.

```
<xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0">
```

### 2.189.13 ScaleRange.StartWidth

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **ScaleRange.StartWidth** element specifies the width of a [ScaleRange](#) at its start, as a percentage of the size of the containing [RadialScale](#) or [LinearScale](#) element. This element is optional.

If the **ScaleRange.StartWidth** element is present, its value MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0. [<52>](#) If this element is not present, its value is interpreted as 0.

The following is the parent element of the **ScaleRange.StartWidth** element.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.StartWidth** element.

```
<xsd:element name="StartWidth" type="xsd:string" minOccurs="0">
```

### 2.189.14 ScaleRange.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleRange.Style** element specifies style properties for a [ScaleRange](#) instance. The **ScaleRange.Style** element is optional. This element is of type [Style](#).

The following is the parent element of the **ScaleRange.Style** element.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0">
```

### 2.189.15 ScaleRange.ToolTip

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ScaleRange.ToolTip** element specifies the tooltip text for a [ScaleRange](#) instance. The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **ScaleRange.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

The following are the parent elements of the **ScaleRange.ToolTip** element.

Parent elements
ScaleRange

The following is the XML Schema definition of the **ScaleRange.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0">
```

### 2.190 TopImage

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TopImage** element specifies an image to be displayed on top of a [LinearGauge](#) or a [RadialGauge](#). This element is optional.

The following are the parent and child elements of the **TopImage** element.

Parent elements
<a href="#">GaugePanel</a>
LinearGauge
<a href="#">TickMarkStyle</a>
<a href="#">GaugeTickMarks</a>
<a href="#">ScalePin</a>
RadialGauge

Child elements
<a href="#">TopImage.MIMETYPE</a>
<a href="#">TopImage.Source</a>
<a href="#">TopImage.TransparentColor</a>
<a href="#">TopImage.Value</a>
<a href="#">TopImage.HueColor</a>

The following is the XML Schema definition of the **TopImage** element in RDL 2008/01.

```
<xsd:complexType name="TopImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **TopImage** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="TopImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.190.1 TopImage.MIMETYPE

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TopImage.MIMETYPE** element specifies the image format of a [TopImage](#) instance. The **TopImage.MIMETYPE** element is optional. If this element is present, its value MUST be a [ReportMIMETYPE](#). If the peer element [TopImage.Source](#) is set to a value other than "Database", the **TopImage.MIMETYPE** element is ignored.

The following is the parent element of the **TopImage.MIMETYPE** element.

Parent elements
TopImage

The following is the XML Schema definition of the **TopImage.MIMETYPE** element.

```
<xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0">
```

## 2.190.2 TopImage.Source

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TopImage.Source** element specifies the type of source that is associated with a [TopImage](#) instance. The **TopImage.Source** element MUST be specified. The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**External:** Specifies that the peer [TopImage.Value](#) element contains a **String** constant or expression that evaluates to the location of an image.

**Embedded:** Specifies that the peer **TopImage.Value** element contains a **String** constant or expression that evaluates to the name of an [EmbeddedImage](#) instance within the report.

**Database:** Specifies that the peer **TopImage.Value** element contains an expression (for example, a field in the database) that evaluates to the binary data for an image.

The following is the parent element of the **TopImage.Source** element.

Parent elements
TopImage

The following is the XML Schema definition of the **TopImage.Source** element.

```
<xsd:element name="Source" type="xsd:string" minOccurs="1">
```

## 2.190.3 TopImage.TransparentColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **TopImage.TransparentColor** element specifies the color to treat as transparent in a [TopImage](#) instance. The **TopImage.TransparentColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) value or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **TopImage.TransparentColor** element.

Parent elements
TopImage

The following is the XML Schema definition of the **TopImage.TransparentColor** element.

```
<xsd:element name="TransparentColor" type="xsd:string" minOccurs="0">
```

## 2.190.4 TopImage.Value

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TopImage.Value** element specifies either the location or the actual data of an image, depending on the value of the peer [TopImage.Source](#) element. The **TopImage.Value** element MUST be specified.

If the peer **TopImage.Source** element is set to "External" and the value of **TopImage.Value** is non-empty, then the value of **TopImage.Value** MUST be a constant [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) value or an expression that evaluates to the location of an image. This location MUST be a [ReportPath](#) or [RdlURL](#) value.

If the peer **TopImage.Source** element is set to "Embedded" and the value of **TopImage.Value** is non-empty, then the value of **TopImage.Value** MUST be a constant **String** value or an expression that evaluates to the name of an [EmbeddedImage](#) in the report.

If the peer **TopImage.Source** element is set to "Database" and the value of **TopImage.Value** is non-empty, then the value of **TopImage.Value** MUST be an expression that evaluates to the binary data for an image.

If this element has an empty value, the image MUST NOT be displayed.

The following is the parent element of the **TopImage.Value** element.

Parent elements
<a href="#">TopImage</a>

The following is the XML Schema definition of the **TopImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1">
```

## 2.190.5 TopImage.HueColor

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **TopImage.HueColor** element specifies the color with which to tint a [TopImage](#). The **TopImage.HueColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **TopImage.HueColor** element.

Parent elements
TopImage

The following is the XML Schema definition of the **TopImage.HueColor** element.

```
<xsd:element name="HueColor" type="xsd:string" minOccurs="0">
```

## 2.191 NumericIndicators

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **NumericIndicators** element is ignored.

The following are the parent and child elements of the **NumericIndicators** element.

Parent elements
<a href="#">GaugePanel</a>

Child elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicators** element in RDL 2008/01.

```
<xsd:complexType name="NumericIndicatorsType">
  <xsd:sequence>
    <xsd:element name="NumericIndicator" type="NumericIndicatorType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **NumericIndicators** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="NumericIndicatorsType">
  <xsd:sequence>
    <xsd:element name="NumericIndicator" type="NumericIndicatorType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.191.1 NumericIndicators.NumericIndicator

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **NumericIndicators.NumericIndicator** element is ignored.

The following is the parent element of the **NumericIndicators.NumericIndicator** element.

Parent elements
<a href="#">NumericIndicators</a>

The following is the XML Schema definition of the **NumericIndicators.NumericIndicator** element.

```
<xsd:element name="NumericIndicator" type="NumericIndicatorType" minOccurs="1"
```

maxOccurs="unbounded" />

## 2.192 NumericIndicator

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **NumericIndicator** element is ignored.

The following are the parent elements, attributes, and child elements of the **NumericIndicator** element.

Parent elements
<a href="#">NumericIndicators</a>

Attributes
<a href="#">NumericIndicator.Name</a>

Child elements
<a href="#">NumericIndicator.ActionInfo</a>
<a href="#">NumericIndicator.Height</a>
<a href="#">NumericIndicator.Hidden</a>
<a href="#">NumericIndicator.Left</a>
<a href="#">NumericIndicator.ParentItem</a>
<a href="#">NumericIndicator.ToolTip</a>
<a href="#">NumericIndicator.Top</a>
<a href="#">NumericIndicator.Width</a>
<a href="#">NumericIndicator.ZIndex</a>
<a href="#">NumericIndicator.DecimalDigitColor</a>
<a href="#">NumericIndicator.DecimalDigits</a>
<a href="#">NumericIndicator.DigitColor</a>
<a href="#">NumericIndicator.Digits</a>
<a href="#">NumericIndicator.GaugeInputValue</a>
<a href="#">NumericIndicator.IndicatorStyle</a>
<a href="#">NumericIndicator.LedDimColor</a>
<a href="#">NumericIndicator.MaximumValue</a>
<a href="#">NumericIndicator.MinimumValue</a>

Child elements
<a href="#">NumericIndicator.Multiplier</a>
<a href="#">NumericIndicator.NumericIndicatorRanges</a>
<a href="#">NumericIndicator.Offstring</a>
<a href="#">NumericIndicator.OutOfRangeString</a>
<a href="#">NumericIndicator.ResizeMode</a>
<a href="#">NumericIndicator.SeparatorColor</a>
<a href="#">NumericIndicator.SeparatorWidth</a>
<a href="#">NumericIndicator.ShowDecimalPoint</a>
<a href="#">NumericIndicator.ShowLeadingZeros</a>
<a href="#">NumericIndicator.ShowSign</a>
<a href="#">NumericIndicator.SnappingEnabled</a>
<a href="#">NumericIndicator.SnappingInterval</a>
<a href="#">NumericIndicator.Style</a>
<a href="#">NumericIndicator.UseFontPercent</a>

The following is the XML Schema definition of the **NumericIndicator** element in RDL 2008/01.

```

<xsd:complexType name="NumericIndicatorType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType"
      minOccurs="1" />
    <xsd:element name="MaximumValue" type="GaugeInputValueType"
      minOccurs="0" />
    <xsd:element name="MinimumValue" type="GaugeInputValueType"
      minOccurs="0" />
    <xsd:element name="NumericIndicatorRanges"
      type="NumericIndicatorRangesType" minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="DecimalDigitColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="DecimalDigits" type="xsd:string" minOccurs="0" />
    <xsd:element name="DigitColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Digits" type="xsd:string" minOccurs="0" />
    <xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="LedDimColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffString" type="xsd:string" minOccurs="0" />
    <xsd:element name="OutOfRangeString" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeparatorWidth" type="xsd:string" minOccurs="0" />
  
```



```

    <xsd:element name="ShowDecimalPoint" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowLeadingZeros" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowSign" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **NumericIndicator** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="NumericIndicatorType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType"
      minOccurs="1" />
    <xsd:element name="MaximumValue" type="GaugeInputValueType"
      minOccurs="0" />
    <xsd:element name="MinimumValue" type="GaugeInputValueType"
      minOccurs="0" />
    <xsd:element name="NumericIndicatorRanges"
      type="NumericIndicatorRangesType" minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="DecimalDigitColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="DecimalDigits" type="xsd:string" minOccurs="0" />
    <xsd:element name="DigitColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Digits" type="xsd:string" minOccurs="0" />
    <xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="LedDimColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffString" type="xsd:string" minOccurs="0" />
    <xsd:element name="OutOfRangeString" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeparatorWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowDecimalPoint" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowLeadingZeros" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowSign" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

## 2.192.1 NumericIndicator.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **NumericIndicator.Name** attribute specifies a unique identifier for a [NumericIndicator](#). The **NumericIndicator.Name** attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [UTR15].

The following is the parent element of the **NumericIndicator.Name** attribute.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.192.2 NumericIndicator.ActionInfo

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.ActionInfo** element is ignored. This element is of type [ActionInfo](#).

The following is the parent element of the **NumericIndicator.ActionInfo** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.192.3 NumericIndicator.Height

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.Height** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**.

The following is the parent element of the **NumericIndicator.Height** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.Height** element.

```
<xsd:element name="Height" type="xsd:string" minOccurs="0" />
```

### 2.192.4 NumericIndicator.Hidden

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.Hidden** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**.

The following is the parent element of the **NumericIndicator.Hidden** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

## 2.192.5 NumericIndicator.Left

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.Left** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**.

The following is the parent element of the **NumericIndicator.Left** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

## 2.192.6 NumericIndicator.ParentItem

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.ParentItem** element is ignored if it is present. However, its data type is validated, and the value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **NumericIndicator.ParentItem** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.ParentItem** element.

```
<xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
```

## 2.192.7 NumericIndicator.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.ToolTip** element is ignored. If it is present, however, its data type is validated, and the value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **NumericIndicator.ToolTip** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

## 2.192.8 NumericIndicator.Top

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.Top** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**.

The following is the parent element of the **NumericIndicator.Top** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.Top** element.

```
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
```

## 2.192.9 NumericIndicator.Width

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.Width** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**.

The following is the parent element of the **NumericIndicator.Width** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
```

## 2.192.10 NumericIndicator.ZIndex

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.ZIndex** element is ignored if it is present. However, its data type is validated and the value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**.

The following is the parent element of the **NumericIndicator.ZIndex** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
```

### 2.192.11 NumericIndicator.DecimalDigitColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.DecimalDigitColor** element is ignored if it is present. However, its data type is validated, and the value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

Following is the parent element of the **NumericIndicator.DecimalDigitColor** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.DecimalDigitColor** element.

```
<xsd:element name="DecimalDigitColor" type="xsd:string" minOccurs="0" />
```

### 2.192.12 NumericIndicator.DecimalDigits

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.DecimalDigits** element is ignored if it is present. However, its data type is validated, and the value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**.

The following is the parent element of the **NumericIndicator.DecimalDigits** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.DecimalDigits** element.

```
<xsd:element name="DecimalDigits" type="xsd:string" minOccurs="0" />
```

### 2.192.13 NumericIndicator.DigitColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.DigitColor** element is ignored if it is present. However, its data type is validated, and the value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **NumericIndicator.DigitColor** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.DigitColor** element.

```
<xsd:element name="DigitColor" type="xsd:string" minOccurs="0" />
```

## 2.192.14 NumericIndicator.Digits

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.Digits** element is ignored if it is present. However, its data type is validated, and the value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**.

The following is the parent element of the **NumericIndicator.Digits** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.Digits** element.

```
<xsd:element name="Digits" type="xsd:string" minOccurs="0" />
```

## 2.192.15 NumericIndicator.GaugeInputValue

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.GaugeInputValue** element is ignored. This element is of type [GaugeInputValue](#).

The following is the parent element of the **NumericIndicator.GaugeInputValue** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.GaugeInputValue** element.

```
<xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="1" />
```

## 2.192.16 NumericIndicator.IndicatorStyle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.IndicatorStyle** element is ignored if it is present. However, its data type is validated and this value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that

evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

- **Mechanical**
- **Digital7Segment**
- **Digital14Segment**

The following is the parent element of the **NumericIndicator.IndicatorStyle** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.IndicatorStyle** element.

```
<xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="0" />
```

### 2.192.17 **NumericIndicator.LedDimColor**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.LedDimColor** element is ignored if it is present. However, its data type is validated, and the value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **NumericIndicator.LedDimColor** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.LedDimColor** element.

```
<xsd:element name="LedDimColor" type="xsd:string" minOccurs="0" />
```

### 2.192.18 **NumericIndicator.MaximumValue**

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.MaximumValue** element is ignored. This element is of type [GaugeInputValue](#).

The following is the parent element of the **NumericIndicator.MaximumValue** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.MaximumValue** element.

```
<xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
```

## 2.192.19 NumericIndicator.MinimumValue

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **NumericIndicator.MinimumValue** element is ignored. This element is of type [GaugeInputValue](#).

The following is the parent element of the **NumericIndicator.MinimumValue** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.MinimumValue** element.

```
<xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
```

## 2.192.20 NumericIndicator.Multiplier

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **NumericIndicator.Multiplier** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**.

The following is the parent element of the **NumericIndicator.Multiplier** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.Multiplier** element.

```
<xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
```

## 2.192.21 NumericIndicator.NumericIndicatorRanges

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **NumericIndicator.NumericIndicatorRanges** element is ignored. This element is of type [NumericIndicatorRanges](#).

The following is the parent element of the **NumericIndicator.NumericIndicatorRanges** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.NumericIndicatorRanges** element.

```
<xsd:element name="NumericIndicatorRanges" type="NumericIndicatorRangesType" minOccurs="0" />
```



## 2.192.22 NumericIndicator.Offstring

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.Offstring** element is ignored.

The following is the parent element of the **NumericIndicator.Offstring** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.Offstring** element.

```
<xsd:element name="OffString" type="xsd:string" minOccurs="0" />
```

## 2.192.23 NumericIndicator.OutOfRangeString

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.OutOfRangeString** element is ignored.

The following is the parent element of the **NumericIndicator.OutOfRangeString** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.OutOfRangeString** element.

```
<xsd:element name="OutOfRangeString" type="xsd:string" minOccurs="0" />
```

## 2.192.24 NumericIndicator.ResizeMode

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.ResizeMode** element is ignored.

The following is the parent element of the **NumericIndicator.ResizeMode** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.ResizeMode** element.

```
<xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
```

## 2.192.25 NumericIndicator.SeparatorColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.SeparatorColor** element is ignored if it is present. However, its data type is validated, and the value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **NumericIndicator.SeparatorColor** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.SeparatorColor** element.

```
<xsd:element name="SeparatorColor" type="xsd:string" minOccurs="0" />
```

## 2.192.26 NumericIndicator.SeparatorWidth

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.SeparatorWidth** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**.

The following is the parent element of the **NumericIndicator.SeparatorWidth** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.SeparatorWidth** element.

```
<xsd:element name="SeparatorWidth" type="xsd:string" minOccurs="0" />
```

## 2.192.27 NumericIndicator.ShowDecimalPoint

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.ShowDecimalPoint** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**.

The following is the parent element of the **NumericIndicator.ShowDecimalPoint** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.ShowDecimalPoint** element.

```
<xsd:element name="ShowDecimalPoint" type="xsd:string" minOccurs="0" />
```

## 2.192.28 NumericIndicator.ShowLeadingZeros

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.ShowLeadingZeros** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**.

The following is the parent element of the **NumericIndicator.ShowLeadingZeros** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.ShowLeadingZeros** element.

```
<xsd:element name="ShowLeadingZeros" type="xsd:string" minOccurs="0" />
```

### 2.192.29 NumericIndicator.ShowSign

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.ShowSign** element is ignored if it is present. However, its data type is validated, and the value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. If the value is not an expression, the value of this element MUST be one of the following:

- NegativeOnly
- Both
- None

The following is the parent element of the **NumericIndicator.ShowSign** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.ShowSign** element.

```
<xsd:element name="ShowSign" type="xsd:string" minOccurs="0" />
```

### 2.192.30 NumericIndicator.SnappingEnabled

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.SnappingEnabled** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**.

The following is the parent element of the **NumericIndicator.SnappingEnabled** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.SnappingEnabled** element.

```
<xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
```

### 2.192.31 NumericIndicator.SnappingInterval

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.SnappingInterval** element is ignored if it is present. However, its data type is validated, and the value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**.

The following is the parent element of the **NumericIndicator.SnappingInterval** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.SnappingInterval** element.

```
<xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
```

### 2.192.32 NumericIndicator.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.Style** element is ignored.

The following is the parent element of the **NumericIndicator.Style** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.192.33 NumericIndicator.UseFontPercent

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicator.UseFontPercent** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**.

The following is the parent element of the **NumericIndicator.UseFontPercent** element.

Parent elements
<a href="#">NumericIndicator</a>

The following is the XML Schema definition of the **NumericIndicator.UseFontPercent** element.

```
<xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
```

## 2.193 NumericIndicatorRanges

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **NumericIndicatorRanges** element is ignored.

The following are the parent and child elements of the **NumericIndicatorRanges** element.

Parent elements
<a href="#">NumericIndicator</a>

Child elements
<a href="#">NumericIndicatorRanges.NumericIndicatorRange</a>

The following is the XML Schema definition of the **NumericIndicatorRanges** element in RDL 2008/01.

```
<xsd:complexType name="NumericIndicatorRangesType">
  <xsd:sequence>
    <xsd:element name="NumericIndicatorRange" type="NumericIndicatorRangeType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **NumericIndicatorRanges** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="NumericIndicatorRangesType">
  <xsd:sequence>
    <xsd:element name="NumericIndicatorRange" type="NumericIndicatorRangeType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.193.1 NumericIndicatorRanges.NumericIndicatorRange

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **NumericIndicatorRanges.NumericIndicatorRange** element is ignored.

Following is the parent element of the **NumericIndicatorRanges.NumericIndicatorRange** element.

Parent elements
<a href="#">NumericIndicatorRanges</a>

The following is the XML Schema definition of the **NumericIndicatorRanges.NumericIndicatorRange** element.

```
<xsd:element name="NumericIndicatorRange" type="NumericIndicatorRangeType"
```

```
minOccurs="1" maxOccurs="unbounded" />
```

## 2.194 NumericIndicatorRange

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **NumericIndicatorRange** element is ignored.

The following are the parent elements, attributes, and child elements of the **NumericIndicatorRange** element.

Parent elements
<a href="#">NumericIndicatorRanges</a>

Attributes
<a href="#">NumericIndicatorRange.Name</a>

Child elements
<a href="#">NumericIndicatorRange.StartValue</a>
<a href="#">NumericIndicatorRange.EndValue</a>
<a href="#">NumericIndicatorRange.DecimalDigitColor</a>
<a href="#">NumericIndicatorRange.DigitColor</a>

The following is the XML Schema definition of the **NumericIndicatorRange** element in RDL 2008/01.

```
<xsd:complexType name="NumericIndicatorRangeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="DecimalDigitColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="DigitColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **NumericIndicatorRange** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="NumericIndicatorRangeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="DecimalDigitColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="DigitColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
```

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.194.1 NumericIndicatorRange.Name

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicatorRange.Name** attribute specifies a unique identifier for a [NumericIndicatorRange](#). The **NumericIndicatorRange.Name** attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [[UTR15](#)].

The following is the parent element of the **NumericIndicatorRange.Name** attribute.

Parent elements
<a href="#">NumericIndicatorRange</a>

The following is the XML Schema definition of the **NumericIndicatorRange.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.194.2 NumericIndicatorRange.StartValue

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicatorRange.StartValue** element is ignored. This element is of type [GaugeInputValue](#).

The following is the parent element of the **NumericIndicatorRange.StartValue** element.

Parent elements
<a href="#">NumericIndicatorRange</a>

The following is the XML Schema definition of the **NumericIndicatorRange.StartValue** element.

```
<xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
```

### 2.194.3 NumericIndicatorRange.EndValue

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicatorRange.EndValue** element is of type [GaugeInputValue](#). This element is ignored.

The following is the parent element of the **NumericIndicatorRange.EndValue** element.

Parent elements
<a href="#">NumericIndicatorRange</a>

The following is the XML Schema definition of the **NumericIndicatorRange.EndValue** element.

```
<xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
```

#### 2.194.4 NumericIndicatorRange.DecimalDigitColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicatorRange.DecimalDigitColor** element is ignored if it is present. However, its data type is validated and the value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **NumericIndicatorRange.DecimalDigitColor** element.

Parent elements
<a href="#">NumericIndicatorRange</a>

The following is the XML Schema definition of the **NumericIndicatorRange.DecimalDigitColor** element.

```
<xsd:element name="DecimalDigitColor" type="xsd:string" minOccurs="0" />
```

#### 2.194.5 NumericIndicatorRange.DigitColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **NumericIndicatorRange.DigitColor** element is ignored if it is present. However, its data type is validated and the value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **NumericIndicatorRange.DigitColor** element.

Parent elements
<a href="#">NumericIndicatorRange</a>

The following is the XML Schema definition of the **NumericIndicatorRange.DigitColor** element.

```
<xsd:element name="DigitColor" type="xsd:string" minOccurs="0" />
```

#### 2.195 RadialGauges

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialGauges** element specifies the set of [RadialGauge](#) instances for a [GaugePanel](#). The **RadialGauges** element is optional. This element MUST contain at least one [RadialGauges.RadialGauge](#) instance.

The following are the parent and child elements of the **RadialGauges** element.

Parent elements
GaugePanel



<b>Child elements</b>
-----------------------

RadialGauges.RadialGauge
--------------------------

The following is the XML Schema definition of the **RadialGauges** element in RDL 2008/01.

```
<xsd:complexType name="RadialGaugesType">
  <xsd:sequence>
    <xsd:element name="RadialGauge" type="RadialGaugeType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **RadialGauges** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="RadialGaugesType">
  <xsd:sequence>
    <xsd:element name="RadialGauge" type="RadialGaugeType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.195.1 RadialGauges.RadialGauge

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialGauges.RadialGauge** element specifies a [RadialGauge](#) within the collection of [RadialGauges](#) for a [GaugePanel](#). The **RadialGauges.RadialGauge** element MUST be specified at least once within a **RadialGauges** collection. This element is of type **RadialGauge**.

The following is the parent element of the **RadialGauges.RadialGauge** element.

<b>Parent elements</b>
------------------------

RadialGauges
--------------

The following is the XML Schema definition of the **RadialGauges.RadialGauge** element.

```
<xsd:element name="RadialGauge" type="RadialGaugeType" minOccurs="1"
  maxOccurs="unbounded" />
```

## 2.196 RadialGauge

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialGauge** element specifies a **RadialGauge** to be drawn within a [GaugePanel](#) instance. The **RadialGauge** element MUST be specified at least once within a [RadialGauges](#) collection.

The following are the parent elements, attributes, and child elements of the **RadialGauge** element.

<b>Parent elements</b>
------------------------

RadialGauges
--------------

Attributes
<a href="#">RadialGauge.Name</a>

Child elements
<a href="#">RadialGauge.PivotX</a>
<a href="#">RadialGauge.PivotY</a>
<a href="#">RadialGauge.AspectRatio</a>
<a href="#">RadialGauge.BackFrame</a>
<a href="#">RadialGauge.ClipContent</a>
<a href="#">RadialGauge.GaugeScales</a>
<a href="#">RadialGauge.TopImage</a>
<a href="#">RadialGauge.ActionInfo</a>
<a href="#">RadialGauge.Height</a>
<a href="#">RadialGauge.Hidden</a>
<a href="#">RadialGauge.Left</a>
<a href="#">RadialGauge.ParentItem</a>
<a href="#">RadialGauge.ToolTip</a>
<a href="#">RadialGauge.Top</a>
<a href="#">RadialGauge.Width</a>
<a href="#">RadialGauge.ZIndex</a>

The following is the XML Schema definition of the **RadialGauge** element in RDL 2008/01.

```
<xsd:complexType name="RadialGaugeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeTypeStart-->
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
    <xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="ClipContent" type="xsd:string" minOccurs="0" />
    <xsd:element name="AspectRatio" type="xsd:string" minOccurs="0" />
    <!--GaugeTypeEnd-->
    <xsd:element name="GaugeScales" type="RadialScalesType" />
  </choice>
</complexType>
```

```

        minOccurs="0" />
        <xsd:element name="PivotX" type="xsd:string" minOccurs="0" />
        <xsd:element name="PivotY" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **RadialGauge** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="RadialGaugeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeTypeStart-->
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
    <xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="ClipContent" type="xsd:string" minOccurs="0" />
    <xsd:element name="AspectRatio" type="xsd:string" minOccurs="0" />
    <!--GaugeTypeEnd-->
    <xsd:element name="GaugeScales" type="RadialScalesType"
      minOccurs="0" />
    <xsd:element name="PivotX" type="xsd:string" minOccurs="0" />
    <xsd:element name="PivotY" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

## 2.196.1 RadialGauge.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialGauge.Name** attribute specifies a unique identifier for a [RadialGauge](#). This attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique within a particular [RadialGauges](#) collection.

The following is the parent element of the **RadialGauge.Name** attribute.

Parent elements
RadialGauge

The following is the XML Schema definition of the **RadialGauge.Name** attribute.

```

<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />

```

## 2.196.2 RadialGauge.ActionInfo

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialGauge.ActionInfo** element specifies the actions for a [RadialGauge](#). This element is optional. This element is of type [ActionInfo](#).

The following is the parent element of the **RadialGauge.ActionInfo** element.

Parent elements
RadialGauge

The following is the XML Schema definition of the **RadialGauge.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

## 2.196.3 RadialGauge.Height

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialGauge.Height** element specifies the height for a [RadialGauge](#) as a percentage of [RadialGauge.ParentItem](#). If **RadialGauge.ParentItem** is not specified, the value of the **RadialGauge.Height** element is interpreted as relative to the height of the [GaugePanel](#).

The **RadialGauge.Height** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If the [GaugePanel.AutoLayout](#) property for the **RadialGauge.Height** element's parent **GaugePanel** element is set to true, the **RadialGauge.Height** element is ignored. If the **RadialGauge.Height** element is not present, its value is interpreted as 0.

The following is the parent element of the **RadialGauge.Height** element.

Parent elements
RadialGauge

The following is the XML Schema definition of the **RadialGauge.Height** element.

```
<xsd:element name="Height" type="xsd:string" minOccurs="0" />
```

## 2.196.4 RadialGauge.Hidden

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialGauge.Hidden** element specifies whether a [RadialGauge](#) is hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **RadialGauge.Hidden** element.

Parent elements
RadialGauge

The following is the XML Schema definition of the **RadialGauge.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

## 2.196.5 RadialGauge.Left

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialGauge.Left** element specifies the distance from the left as a percentage of [RadialGauge.ParentItem](#). If **RadialGauge.ParentItem** is not specified, the value of the **RadialGauge.Left** element is interpreted relative to the left of the [GaugePanel](#).

The **RadialGauge.Left** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If the [GaugePanel.AutoLayout](#) property for the **RadialGauge.Left** element's parent **GaugePanel** element is set to true, the **RadialGauge.Left** element is ignored. If the **RadialGauge.Left** element is not present, its value is interpreted as 0.

The following is the parent element of the **RadialGauge.Left** element.

Parent elements
<a href="#">RadialGauge</a>

The following is the XML Schema definition of the **RadialGauge.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

## 2.196.6 RadialGauge.ParentItem

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialGauge.ParentItem** element specifies the name of the parent [RadialGauge](#). The **RadialGauge.ParentItem** element is optional.

The following is the parent element of the **RadialGauge.ParentItem** element.

Parent elements
<a href="#">RadialGauge</a>

The following is the XML Schema definition of the **RadialGauge.ParentItem** element.

```
<xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
```

## 2.196.7 RadialGauge.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialGauge.ToolTip** element specifies the tooltip text for a [RadialGauge](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **RadialGauge.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **RadialGauge.ToolTip** element.

Parent elements
RadialGauge

The following is the XML Schema definition of the **RadialGauge.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

## 2.196.8 RadialGauge.Top

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialGauge.Top** element specifies the distance from the top as a percentage of [RadialGauge.ParentItem](#). If **RadialGauge.ParentItem** is not specified, the value of the **RadialGauge.Top** element is interpreted as relative to the top of the [GaugePanel](#).

This element is optional. If the **RadialGauge.Top** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If the [GaugePanel.AutoLayout](#) property for the **RadialGauge.Top** element's parent **GaugePanel** element is set to true, the **RadialGauge.Top** element is ignored.

If the **RadialGauge.Top** element is not present, its value is interpreted as 0.

The following is the parent element of the **RadialGauge.Top** element.

Parent elements
<a href="#">RadialGauge</a>

The following is the XML Schema definition of the **RadialGauge.Top** element.

```
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
```

## 2.196.9 RadialGauge.Width

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialGauge.Width** element specifies the width of a [RadialGauge](#) as a percentage of [RadialGauge.ParentItem](#). If **RadialGauge.ParentItem** is not specified, the value of the **RadialGauge.Width** element is interpreted as relative to the width of the [GaugePanel](#).

This element is optional. If the **RadialGauge.Width** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If the [GaugePanel.AutoLayout](#) property for the **RadialGauge.Width** element's parent **GaugePanel** element is set to true, the **RadialGauge.Width** element is ignored.

If the **RadialGauge.Width** element is not present, its value is interpreted as 0.

The following is the parent element of the **RadialGauge.Width** element.

Parent elements
RadialGauge

The following is the XML Schema definition of the **RadialGauge.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
```

### 2.196.10 RadialGauge.ZIndex

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialGauge.ZIndex** element specifies the drawing order of a [RadialGauge](#) within a [GaugePanel](#). This element is optional. If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 0. The value of this element MUST greater than or equal to 0 and less than or equal to 2147483647.

The following is the parent element of the **RadialGauge.ZIndex** element.

Parent elements
RadialGauge

The following is the XML Schema definition of the **RadialGauge.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
```

### 2.196.11 RadialGauge.AspectRatio

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialGauge.AspectRatio** element specifies the ratio of the height to the width for drawing a [RadialGauge](#). This element is optional.

If the **RadialGauge.AspectRatio** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of the **RadialGauge.AspectRatio** element MUST be greater than or equal to 0. If this element is not present, its value is interpreted as 0. If the value of this element is 0 or if this element is not present, its value is interpreted based on the content of the **gauge**.

The following is the parent element of the **RadialGauge.AspectRatio** element.

Parent elements
RadialGauge

The following is the XML Schema definition of the **RadialGauge.AspectRatio** element.

```
<xsd:element name="AspectRatio" type="xsd:string" minOccurs="0" />
```

### 2.196.12 RadialGauge.BackFrame

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialGauge.BackFrame** element specifies the background or frame for a [RadialGauge](#). This element is optional. This element is of type [BackFrame](#).

The following is the parent element of the **RadialGauge.BackFrame** element.

Parent elements
RadialGauge

The following is the XML Schema definition of the **RadialGauge.BackFrame** element.

```
<xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
```

### 2.196.13 RadialGauge.ClipContent

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialGauge.ClipContent** element indicates whether the content of a [RadialGauge](#) is clipped by the bounds or frame of the **radial gauge**. This element is optional. If this element is present, its value **MUST** be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **RadialGauge.ClipContent** element.

Parent elements
RadialGauge

The following is the XML Schema definition of the **RadialGauge.ClipContent** element.

```
<xsd:element name="ClipContent" type="xsd:string" minOccurs="0" />
```

### 2.196.14 RadialGauge.GaugeScales

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialGauge.GaugeScales** element specifies the set of scales to display on a [RadialGauge](#). This element is optional. This element is of type [RadialScales](#).

The following is the parent element of the **RadialGauge.GaugeScales** element.

Parent elements
RadialGauge

The following is the XML Schema definition of the **RadialGauge.GaugeScales** element.

```
<xsd:element name="GaugeScales" type="RadialScalesType" minOccurs="0" />
```

### 2.196.15 RadialGauge.TopImage

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialGauge.TopImage** element specifies the image to display over the top of a [RadialGauge](#). This element is optional. This element is of type [TopImage](#).

The following is the parent element of the **RadialGauge.TopImage** element.



<b>Parent elements</b>
<a href="#">RadialGauge</a>

The following is the XML Schema definition of the **RadialGauge.TopImage** element.

```
<xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
```

## 2.196.16 RadialGauge.PivotX

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialGauge.PivotX** element specifies the x-position of the pivot point as a percentage of [RadialGauge.Width](#). The **RadialGauge.PivotX** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 50.

The following is the parent element of the **RadialGauge.PivotX** element.

<b>Parent elements</b>
<a href="#">RadialGauge</a>

The following is the XML Schema definition of the **RadialGauge.PivotX** element.

```
<xsd:element name="PivotX" type="xsd:string" minOccurs="0">
```

## 2.196.17 RadialGauge.PivotY

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialGauge.PivotY** element specifies the y-position of the pivot point as a percentage of [RadialGauge.Height](#) value for a [RadialGauge](#). The **RadialGauge.PivotY** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 50.

The following is the parent element of the **RadialGauge.PivotY** element.

<b>Parent elements</b>
<a href="#">RadialGauge</a>

The following is the XML Schema definition of the **RadialGauge.PivotY** element.

```
<xsd:element name="PivotY" type="xsd:string" minOccurs="0">
```

## 2.197 RadialScales

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialScales** element specifies the set of [RadialScale](#) instances for a [RadialGauge](#). The **RadialScales** element MUST contain at least one [RadialScales.RadialScale](#) instance. This element is optional.

The following are the parent and child elements of the **RadialScales** element.

Parent elements
RadialGauge

Child elements
RadialScales.RadialScale

The following is the XML Schema definition of the **RadialScales** element in RDL 2008/01.

```
<xsd:complexType name="RadialScalesType">
  <xsd:sequence>
    <xsd:element name="RadialScale" type="RadialScaleType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **RadialScales** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="RadialScalesType">
  <xsd:sequence>
    <xsd:element name="RadialScale" type="RadialScaleType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.197.1 RadialScales.RadialScale

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScales.RadialScale** element specifies a [RadialScale](#) within a set of [RadialGauge.GaugeScales](#). The **RadialScales.RadialScale** element is of type **RadialScale**. This element **MUST** be specified at least once within a [RadialScales](#) collection.

The following is the parent element of the **RadialScales.RadialScale** element.

Parent elements
RadialScales

The following is the XML Schema definition of the **RadialScales.RadialScale** element.

```
<xsd:element name="RadialScale" type="RadialScaleType" minOccurs="1"
  maxOccurs="unbounded" />
```

### 2.198 RadialScale

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale** element specifies a **radial scale** to be drawn within a [RadialGauge](#) instance. The **RadialScale** element MUST be specified at least once within a [RadialScales](#) collection.

The following are the parent elements, attributes, and child elements of the **RadialScale** element.

Parent elements
<a href="#">RadialScales</a>

Attributes
<a href="#">RadialScale.Name</a>

Child elements
<a href="#">RadialScale.Radius</a>
<a href="#">RadialScale.StartAngle</a>
<a href="#">RadialScale.SweepAngle</a>
<a href="#">RadialScale.ActionInfo</a>
<a href="#">RadialScale.CustomLabels</a>
<a href="#">RadialScale.GaugeMajorTickMarks</a>
<a href="#">RadialScale.GaugeMinorTickMarks</a>
<a href="#">RadialScale.GaugePointers</a>
<a href="#">RadialScale.Hidden</a>
<a href="#">RadialScale.Interval</a>
<a href="#">RadialScale.IntervalOffset</a>
<a href="#">RadialScale.Logarithmic</a>
<a href="#">RadialScale.LogarithmicBase</a>
<a href="#">RadialScale.MaximumPin</a>
<a href="#">RadialScale.MaximumValue</a>
<a href="#">RadialScale.MinimumPin</a>
<a href="#">RadialScale.MinimumValue</a>
<a href="#">RadialScale.Multiplier</a>
<a href="#">RadialScale.Reversed</a>
<a href="#">RadialScale.ScaleLabels</a>
<a href="#">RadialScale.ScaleRanges</a>
<a href="#">RadialScale.Style</a>

Child elements
<a href="#">RadialScale.TickMarksOnTop</a>
<a href="#">RadialScale.ToolTip</a>
<a href="#">RadialScale.Width</a>

The following is the XML Schema definition of the **RadialScale** element in RDL 2008/01.

```
<xsd:complexType name="RadialScaleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeScaleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="ScaleRanges" type="ScaleRangesType" minOccurs="0" />
    <xsd:element name="ScaleLabels" type="ScaleLabelsType" minOccurs="0" />
    <xsd:element name="GaugeMajorTickMarks" type="GaugeTickMarksType"
      minOccurs="0" />
    <xsd:element name="GaugeMinorTickMarks" type="GaugeTickMarksType"
      minOccurs="0" />
    <xsd:element name="CustomLabels" type="CustomLabelsType" minOccurs="0" />
    <xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MaximumPin" type="ScalePinType" minOccurs="0" />
    <xsd:element name="MinimumPin" type="ScalePinType" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="Logarithmic" type="xsd:string" minOccurs="0" />
    <xsd:element name="LogarithmicBase" type="xsd:string" minOccurs="0" />
    <xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="TickMarksOnTop" type="xsd:string" minOccurs="0" />
    <!--GaugeScaleTypeEnd-->
    <xsd:element name="GaugePointers" type="RadialPointersType" minOccurs="0" />
    <xsd:element name="Radius" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="SweepAngle" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **RadialScale** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="RadialScaleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeScaleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="ScaleRanges" type="ScaleRangesType" minOccurs="0" />
    <xsd:element name="ScaleLabels" type="ScaleLabelsType" minOccurs="0" />
    <xsd:element name="GaugeMajorTickMarks" type="GaugeTickMarksType"
      minOccurs="0" />
    <xsd:element name="GaugeMinorTickMarks" type="GaugeTickMarksType"
      minOccurs="0" />
    <xsd:element name="CustomLabels" type="CustomLabelsType" minOccurs="0" />
    <xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

```

<xsd:element name="MaximumPin" type="ScalePinType" minOccurs="0" />
<xsd:element name="MinimumPin" type="ScalePinType" minOccurs="0" />
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
<xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
<xsd:element name="Logarithmic" type="xsd:string" minOccurs="0" />
<xsd:element name="LogarithmicBase" type="xsd:string" minOccurs="0" />
<xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
<xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
<xsd:element name="TickMarksOnTop" type="xsd:string" minOccurs="0" />
<!--GaugeScaleTypeEnd-->
  <xsd:element name="GaugePointers" type="RadialPointersType" minOccurs="0" />
  <xsd:element name="Radius" type="xsd:string" minOccurs="0" />
  <xsd:element name="StartAngle" type="xsd:string" minOccurs="0" />
  <xsd:element name="SweepAngle" type="xsd:string" minOccurs="0" />
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.198.1 RadialScale.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.Name** attribute specifies the name for a radial scale. This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [UTR15] within a particular [RadialScales](#) collection.

The following is the parent element of the **RadialScale.Name** attribute.

Parent elements
<a href="#">RadialScale</a>

The following is the XML Schema definition of the **RadialScale.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.198.2 RadialScale.ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.ActionInfo** element specifies the actions for a [RadialScale](#). This element is optional. This element is of type [ActionInfo](#).

The following is the parent element of the **RadialScale.ActionInfo** element.

Parent elements
<a href="#">RadialScale</a>

The following is the XML Schema definition of the **RadialScale.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.198.3 RadialScale.CustomLabels

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialScale.CustomLabels** element specifies the custom labels for a [RadialScale](#). This element is optional. This element is of type [CustomLabels](#).

The following is the parent element of the **RadialScale.CustomLabels** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.CustomLabels** element.

```
<xsd:element name="CustomLabels" type="CustomLabelsType" minOccurs="0" />
```

### 2.198.4 RadialScale.GaugeMajorTickMarks

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialScale.GaugeMajorTickMarks** element specifies the major tick marks to display on a [RadialScale](#). This element is optional. This element is of type [GaugeTickMarks](#).

The following is the parent element of the **RadialScale.GaugeMajorTickMarks** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.GaugeMajorTickMarks** element.

```
<xsd:element name="GaugeMajorTickMarks" type="GaugeTickMarksType" minOccurs="0" />
```

### 2.198.5 RadialScale.GaugeMinorTickMarks

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialScale.GaugeMinorTickMarks** element specifies the minor tick marks to display on a [RadialScale](#). This element is optional. This element is of type [GaugeTickMarks](#).

The following is the parent element of the **RadialScale.GaugeMinorTickMarks** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.GaugeMinorTickMarks** element.

```
<xsd:element name="GaugeMinorTickMarks" type="GaugeTickMarksType" minOccurs="0" />
```

## 2.198.6 RadialScale.GaugePointers

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.GaugePointers** element specifies the **radial pointers** to display on a [RadialScale](#). This element is optional. This element is of type [RadialPointers](#).

The following is the parent element of the **RadialScale.GaugePointers** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.GaugePointers** element.

```
<xsd:element name="GaugePointers" type="RadialPointersType" minOccurs="0" />
```

## 2.198.7 RadialScale.Hidden

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.Hidden** element specifies whether a [RadialScale](#) is hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **RadialScale.Hidden** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

## 2.198.8 RadialScale.Interval

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.Interval** element specifies the default interval between [GaugeTickMarks](#) instances and [RadialScale.ScaleLabels](#) instances within a [RadialScale](#). The **RadialScale.Interval** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0 and less than or equal to  $1.79769^{+308}$ .

If this element is not present, its value is interpreted as 0.

The following is the parent element of the **RadialScale.Interval** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.Interval** element.

```
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
```

## 2.198.9 RadialScale.IntervalOffset

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.IntervalOffset** element specifies the default offset for the first [GaugeTickMarks](#) instance and [RadialScale.ScaleLabels](#) instance within a [RadialScale](#). The **RadialScale.IntervalOffset** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be non-negative. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **RadialScale.IntervalOffset** element.

Parent elements
<a href="#">RadialScale</a>

The following is the XML Schema definition of the **RadialScale.IntervalOffset** element.

```
<xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
```

## 2.198.10 RadialScale.Logarithmic

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.Logarithmic** element specifies whether a [RadialScale](#) is logarithmic. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **RadialScale.Logarithmic** element.

Parent elements
<a href="#">RadialGauge</a>

The following is the XML Schema definition of the **RadialScale.Logarithmic** element.

```
<xsd:element name="Logarithmic" type="xsd:string" minOccurs="0" />
```

## 2.198.11 RadialScale.LogarithmicBase

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.LogarithmicBase** element specifies the base to use for a [RadialScale](#) that is logarithmic. This element is optional. If [RadialScale.Logarithmic](#) is set to true, the **RadialScale.LogarithmicBase** element is ignored. If the **RadialScale.LogarithmicBase** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than 1. If this element is not present, its value is interpreted as 10.

The following is the parent element of the **RadialScale.LogarithmicBase** element.



<b>Parent elements</b>
RadialScale

The following is the XML Schema definition of the **RadialScale.LogarithmicBase** element.

```
<xsd:element name="LogarithmicBase" type="xsd:string" minOccurs="0" />
```

## 2.198.12 RadialScale.MaximumPin

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.MaximumPin** element specifies the maximum value at which a [PointerImage](#) on a [RadialScale](#) will stop. This element is optional. This element is of type [ScalePin](#).

The following is the parent element of the **RadialScale.MaximumPin** element.

<b>Parent elements</b>
RadialScale

The following is the XML Schema definition of the **RadialScale.MaximumPin** element.

```
<xsd:element name="MaximumPin" type="ScalePinType" minOccurs="0" />
```

## 2.198.13 RadialScale.MaximumValue

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.MaximumValue** element specifies the maximum value for a [RadialScale](#). This element is optional.

If the **RadialScale.MaximumValue** element is present, the [GaugeInputValue.Value](#) within it MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If **GaugeInputValue.Value** is not set within the **RadialScale.MaximumValue** element, the value of **RadialScale.MaximumValue** is interpreted as 100. The **GaugeInputValue.Value** within the **RadialScale.MaximumValue** element MUST be greater than the **GaugeInputValue.Value** within its peer [RadialScale.MinimumValue](#) element.

The **RadialScale.MaximumValue** element is of type [GaugeInputValue](#).

The following is the parent element of the **RadialScale.MaximumValue** element.

<b>Parent elements</b>
RadialScale

The following is the XML Schema definition of the **RadialScale.MaximumValue** element.

```
<xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
```

## 2.198.14 RadialScale.MinimumPin

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialScale.MinimumPin** element specifies the minimum value at which a [RadialPointer](#) on a [RadialScale](#) will stop. This element is optional. This element is of type [ScalePin](#).

The following is the parent element of the **RadialScale.MinimumPin** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.MinimumPin** element.

```
<xsd:element name="MinimumPin" type="ScalePinType" minOccurs="0" />
```

## 2.198.15 RadialScale.MinimumValue

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialScale.MinimumValue** element specifies the minimum value for a [RadialScale](#). This element is optional.

If the **RadialScale.MinimumValue** element is present, the [GaugeInputValue.Value](#) within it MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to **Float**. If the **GaugeInputValue.Value** is not set within the **RadialScale.MinimumValue** element, the value of **RadialScale.MinimumValue** is interpreted as 0. The **GaugeInputValue.Value** within the **RadialScale.MinimumValue** element MUST be less than the **GaugeInputValue.Value** within its peer [RadialScale.MaximumValue](#) element.

This element is of type [GaugeInputValue](#).

The following is the parent element of the **RadialScale.MinimumValue** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.MinimumValue** element.

```
<xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
```

## 2.198.16 RadialScale.Multiplier

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialScale.Multiplier** element specifies the amount by which values on a [RadialScale](#) are multiplied before they are displayed. This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 1.

The following is the parent element of the **RadialScale.Multiplier** element.

<b>Parent elements</b>
RadialScale

The following is the XML Schema definition of the **RadialScale.Multiplier** element.

```
<xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
```

## 2.198.17 RadialScale.Reversed

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.Reversed** element specifies whether the direction of a [RadialScale](#) is reversed. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **RadialScale.Reversed** element.

<b>Parent elements</b>
RadialScale

The following is the XML Schema definition of the **RadialScale.Reversed** element.

```
<xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
```

## 2.198.18 RadialScale.ScaleLabels

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.ScaleLabels** element specifies labels to display on a [RadialScale](#). This element is optional. This element is of type [ScaleLabels](#).

The following is the parent element of the **RadialScale.ScaleLabels** element.

<b>Parent elements</b>
RadialScale

The following is the XML Schema definition of the **RadialScale.ScaleLabels** element.

```
<xsd:element name="ScaleLabels" type="ScaleLabelsType" minOccurs="0" />
```

## 2.198.19 RadialScale.ScaleRanges

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.ScaleRanges** element specifies ranges to display on a [RadialScale](#). This element is optional. This element is of type [ScaleRanges](#).

The following is the parent element of the **RadialScale.ScaleRanges** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.ScaleRanges** element.

```
<xsd:element name="ScaleRanges" type="ScaleRangesType" minOccurs="0" />
```

## 2.198.20 RadialScale.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.Style** element specifies style properties for a [RadialScale](#). This element is optional. This element is of type [Style](#).

The following is the parent element of the **RadialScale.Style** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.198.21 RadialScale.TickMarksOnTop

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.TickMarksOnTop** element specifies whether instances of [GaugeTickMarks](#) within a [RadialScale](#) are drawn on top of [RadialPointer](#) instances within [RadialScale.GaugePointers](#).

The **RadialScale.TickMarksOnTop** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **RadialScale.TickMarksOnTop** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.TickMarksOnTop** element.

```
<xsd:element name="TickMarksOnTop" type="xsd:string" minOccurs="0" />
```

## 2.198.22 RadialScale.ToolTip

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.ToolTip** element specifies the tooltip text for a [RadialScale](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The

**RadialScale.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **RadialScale.ToolTip** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

### 2.198.23 RadialScale.Width

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.Width** element specifies the width of a [RadialScale](#) as a percentage of the [RadialGauge](#) size. This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **RadialScale.Width** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
```

### 2.198.24 RadialScale.Radius

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialScale.Radius** element specifies the radius of a [RadialScale](#) as a percentage of the size of the parent [RadialGauge](#) element. The **RadialScale.Radius** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST greater than or equal to 0 and less than or equal to  $1.79769+308$ . If the **RadialScale.Radius** element is not present, its value is interpreted as 37.

The following is the parent element of the **RadialScale.Radius** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.Radius** element.

```
<xsd:element name="Radius" type="xsd:string" minOccurs="0">
```

## 2.198.25 RadialScale.StartAngle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialScale.StartAngle** element specifies the start angle of a [RadialScale](#) in terms of degrees. This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0 and less than or equal to 360. If this element is not present, the **Float** value is interpreted as 20.

The following is the parent element of the **RadialScale.StartAngle** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.StartAngle** element.

```
<xsd:element name="StartAngle" type="xsd:string" minOccurs="0">
```

## 2.198.26 RadialScale.SweepAngle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialScale.SweepAngle** element specifies the sweep angle of a [RadialScale](#) in terms of degrees. This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0 and less than or equal to 360. If the **RadialScale.SweepAngle** element is not present, its value is interpreted as 320.

The following is the parent element of the **RadialScale.SweepAngle** element.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.SweepAngle** element.

```
<xsd:element name="SweepAngle" type="xsd:string" minOccurs="0">
```

## 2.199 RadialPointers

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialPointers** element specifies the set of [RadialPointer](#) instances for a [RadialScale](#). The **RadialPointers** element MUST contain at least one [RadialPointers.RadialPointer](#) instance and can contain more.

The following are the parent and child elements of the **RadialPointers** element.

Parent elements
RadialScale

<b>Child elements</b>
-----------------------

RadialPointers.RadialPointer
------------------------------

The following is the XML Schema definition of the **RadialPointers** element.

```
<xsd:complexType name="RadialPointersType">
  <xsd:sequence>
    <xsd:element name="RadialPointer" type="RadialPointerType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

### 2.199.1 RadialPointers.RadialPointer

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialPointers.RadialPointer** element specifies a [RadialPointer](#) within the collection of [RadialPointers](#) for a [RadialScale](#). The **RadialPointers.RadialPointer** element is of type **RadialPointer**.

The following is the parent element of the **RadialPointers.RadialPointer** element. This element MUST be specified at least once.

<b>Parent elements</b>
------------------------

RadialPointers
----------------

The following is the XML Schema definition of the **RadialPointers.RadialPointer** element.

```
<xsd:element name="RadialPointer" type="RadialPointerType" minOccurs="1"
  maxOccurs="unbounded" />
```

### 2.200 RadialPointer

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialPointer** element specifies a **RadialPointer** to be drawn against a [RadialScale](#). The **RadialPointer** element is optional. This element MUST be specified at least once within a [RadialPointers](#) collection.

The following are the parent elements, attributes, and child elements of the **RadialPointer** element.

<b>Parent elements</b>
------------------------

RadialPointers
----------------

<b>Attributes</b>
-------------------

<a href="#">RadialPointer.Name</a>
------------------------------------

Child elements
<a href="#">RadialPointer.ActionInfo</a>
<a href="#">RadialPointer.BarStart</a>
<a href="#">RadialPointer.DistanceFromScale</a>
<a href="#">RadialPointer.GaugeInputValue</a>
<a href="#">RadialPointer.Hidden</a>
<a href="#">RadialPointer.MarkerLength</a>
<a href="#">RadialPointer.MarkerStyle</a>
<a href="#">RadialPointer.Placement</a>
<a href="#">RadialPointer.PointerImage</a>
<a href="#">RadialPointer.SnappingEnabled</a>
<a href="#">RadialPointer.SnappingInterval</a>
<a href="#">RadialPointer.Style</a>
<a href="#">RadialPointer.ToolTip</a>
<a href="#">RadialPointer.Width</a>
<a href="#">RadialPointer.NeedleStyle</a>
<a href="#">RadialPointer.PointerCap</a>
<a href="#">RadialPointer.Type</a>

The following is the XML Schema definition of the **RadialPointer** element in RDL 2008/01.

```

<xsd:complexType name="RadialPointerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePointerTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="PointerImage" type="PointerImageType" minOccurs="0" />
    <xsd:element name="BarStart" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerLength" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <!--GaugePointerTypeEnd-->
    <xsd:element name="PointerCap" type="PointerCapType" minOccurs="0" />
    <xsd:element name="NeedleStyle" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```



The following is the XML Schema definition of the **RadialPointer** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="RadialPointerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePointerTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="PointerImage" type="PointerImageType" minOccurs="0" />
    <xsd:element name="BarStart" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerLength" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <!--GaugePointerTypeEnd-->
    <xsd:element name="PointerCap" type="PointerCapType" minOccurs="0" />
    <xsd:element name="NeedleStyle" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.200.1 RadialPointer.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialPointer.Name** attribute specifies a unique identifier for a [RadialPointer](#). This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique within a particular [RadialPointers](#) collection.

The following is the parent element of the **RadialPointer.Name** attribute.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

## 2.200.2 RadialPointer.ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialPointer.ActionInfo** element specifies the actions for a [RadialPointer](#). This element is optional. This element is of type [ActionInfo](#).

The following is the parent element of the **RadialPointer.ActionInfo** element.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.200.3 RadialPointer.BarStart

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialPointer.BarStart** element specifies where the pointer will start for a [RadialPointer](#) if the **radial pointer** is of type "Bar". This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. This element is ignored if the associated [RadialPointer.Type](#) is not set to "Bar". The value of the **RadialPointer.BarStart** element MUST be one of the following:

**ScaleStart:** The bar starts from the starting point of the [RadialScale](#).

**Zero:** The bar starts from the zero point of the **RadialScale**.

If the **RadialPointer.BarStart** element is not present, its value is interpreted as "ScaleStart".

The following is the parent element of the **RadialPointer.BarStart** element.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.BarStart** element.

```
<xsd:element name="BarStart" type="xsd:string" minOccurs="0" />
```

### 2.200.4 RadialPointer.DistanceFromScale

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialPointer.DistanceFromScale** element specifies the distance from the tip of the [RadialPointer](#) to its associated [RadialScale](#) as a percentage of the size of the radial scale.

The **RadialPointer.DistanceFromScale** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a [Float.<53>](#) If the **RadialPointer.DistanceFromScale** element is not present, its value is interpreted as 0.

The following is the parent element of the **RadialPointer.DistanceFromScale** element.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.DistanceFromScale** element.

```
<xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
```

## 2.200.5 RadialPointer.GaugeInputValue

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialPointer.GaugeInputValue** element specifies the value to use for a [RadialPointer](#). This element is optional. This element is of type [GaugeInputValue](#).

The following is the parent element of the **RadialPointer.GaugeInputValue** element.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.GaugeInputValue** element.

```
<xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="0" />
```

## 2.200.6 RadialPointer.Hidden

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialPointer.Hidden** element specifies whether a [RadialPointer](#) is hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **RadialPointer.Hidden** element.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

## 2.200.7 RadialPointer.MarkerLength

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialPointer.MarkerLength** element specifies the length of the marker for a [RadialPointer](#) as a percentage of the parent [RadialScale](#) radius. The **RadialPointer.MarkerLength** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0. [<54>](#) If the **RadialPointer.MarkerLength** element is not present, its value is interpreted as 0.

The following is the parent element of the **RadialPointer.MarkerLength** element.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.MarkerLength** element.

```
<xsd:element name="MarkerLength" type="xsd:string" minOccurs="0" />
```

## 2.200.8 RadialPointer.MarkerStyle

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialPointer.MarkerStyle** element specifies the type of the marker for a [RadialPointer](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Triangle:** The marker is of type "Triangle".

**Rectangle:** The marker is of type "Rectangle".

**Circle:** The marker is of type "Circle".

**Diamond:** The marker is of type "Diamond".

**Trapezoid:** The marker is of type "Trapezoid".

**Star:** The marker is of type "Star".

**Wedge:** The marker is of type "Wedge".

**Pentagon:** The marker is of type "Pentagon".

**None:** The marker is of type "None".

If the **RadialPointer.MarkerStyle** element is not present, its value is interpreted as "Triangle".

The following is the parent element of the **RadialPointer.MarkerStyle** element.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.MarkerStyle** element.

```
<xsd:element name="MarkerStyle" type="xsd:string" minOccurs="0" />
```

## 2.200.9 RadialPointer.Placement

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **RadialPointer.Placement** element determines where a [RadialPointer](#) is placed relative to a [RadialScale](#). The **RadialPointer.Placement** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value for this element MUST be one of the following:

**Inside:** The pointer is placed inside the radial scale.

**Outside:** The pointer is placed outside the radial scale.

**Cross:** The pointer is placed across the radial scale.

If the **RadialPointer.Placement** element is not present, its value is interpreted as "Inside".

The following is the parent element of the **RadialPointer.Placement** element.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.Placement** element.

```
<xsd:element name="Placement" type="xsd:string" minOccurs="0" />
```

### 2.200.10 RadialPointer.PointerImage

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialPointer.PointerImage** element specifies the image to use for a **radial pointer**. This element is optional. This element is of type [PointerImage](#).

The following is the parent element of the **RadialPointer.PointerImage** element.

Parent elements
<a href="#">RadialPointer</a>

The following is the XML Schema definition of the **RadialPointer.PointerImage** element.

```
<xsd:element name="PointerImage" type="PointerImageType" minOccurs="0" />
```

### 2.200.11 RadialPointer.SnappingEnabled

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialPointer.SnappingEnabled** element specifies whether values within a [RadialPointer](#) round to the snapping interval. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **RadialPointer.SnappingEnabled** element.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.SnappingEnabled** element.

```
<xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
```

### 2.200.12 RadialPointer.SnappingInterval

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialPointer.SnappingInterval** element specifies the interval to which values within a [RadialPointer](#) round. The **RadialPointer.SnappingInterval** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of the **RadialPointer.SnappingInterval** element MUST be non-negative. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **RadialPointer.SnappingInterval** element.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.SnappingInterval** element.

```
<xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
```

### 2.200.13 RadialPointer.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialPointer.Style** element specifies style properties for a [RadialPointer](#). This element is optional. This element is of type [Style](#).

The following is the parent element of the **RadialPointer.Style** element.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.200.14 RadialPointer.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialPointer.ToolTip** element specifies the tooltip text for a [RadialPointer](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **RadialPointer.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **RadialPointer.ToolTip** element.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.ToolTip** element.

```
<xsd:element name="Tooltip" type="xsd:string" minOccurs="0" />
```

## 2.200.15 RadialPointer.Width

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialPointer.Width** element specifies the width of a [RadialPointer](#) as a percentage of [RadialScale.Width](#). The **RadialPointer.Width** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **RadialPointer.Width** element.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
```

## 2.200.16 RadialPointer.NeedleStyle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialPointer.NeedleStyle** element specifies the style of a **radial pointer** needle. This element is optional. This element is ignored if peer element [RadialPointer.Type](#) is not set to "Needle".

If the **RadialPointer.NeedleStyle** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Triangular:** Specifies that the needle is shaped like a triangle.

**Rectangular:** Specifies that the needle is shaped like a rectangle.

**TaperedWithTail:** Specifies that the needle is tapered and contains a rectangular tail.

**Tapered:** Specifies that the needle is tapered without a rectangular tail.

**ArrowWithTail:** Specifies that the needle is shaped like an arrow with a rectangular tail.

**Arrow:** Specifies that the needle is shaped like an arrow without a rectangular tail.

**StealthArrowWithTail:** Specifies that the needle is shaped like an arrow with a diagonal tip with a rectangular tail.

**StealthArrow:** Specifies that the needle is shaped like an arrow with a diagonal tip without a rectangular tail.

**TaperedWithStealthArrow:** Specifies that the needle is tapered and has a stealth arrow tip.

**StealthArrowWithWideTail:** Specifies that the needle has a stealth arrow tip and a diagonally widening tail.

**TaperedWithRoundedPoint:** Specifies that the needle is tapered with a rounded end.

If the **RadialPointer.NeedleStyle** element is not present, its value is interpreted as "Triangular".

The following is the parent element of the **RadialPointer.NeedleStyle** element.

<b>Parent elements</b>
------------------------

<a href="#">RadialPointer</a>
-------------------------------

The following is the XML Schema definition of the **RadialPointer.NeedleStyle** element.

```
<xsd:element name="NeedleStyle" type="xsd:string" minOccurs="0">
```

## 2.200.17 RadialPointer.PointerCap

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialPointer.PointerCap** element specifies style properties for the pointer cap of a [RadialPointer](#). This element is optional. This element is of type [PointerCap](#).

The following is the parent element of the **RadialPointer.PointerCap** element.

<b>Parent elements</b>
------------------------

<a href="#">RadialPointer</a>
-------------------------------

The following is the XML Schema definition of the **RadialPointer.PointerCap** element.

```
<xsd:element name="PointerCap" type="PointerCapType" minOccurs="0">
```

## 2.200.18 RadialPointer.Type

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **RadialPointer.Type** element specifies the type of a [RadialPointer](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Needle:** Specifies that the radial pointer is a needle.

**Marker:** Specifies that the radial pointer is a marker.

**Bar:** Specifies that the radial pointer is a bar.

If the **RadialPointer.Type** element is not present, its value is interpreted as "Needle".

The following is the parent element of the **RadialPointer.Type** element.

<b>Parent elements</b>
------------------------

<a href="#">RadialPointer</a>
-------------------------------

The following is the XML Schema definition of the **RadialPointer.Type** element.

```
<xsd:element name="Type" type="xsd:string" minOccurs="0">
```



## 2.201 PointerCap

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PointerCap** element specifies display properties for the [RadialPointer](#) caps within a [RadialGauge](#). This element is optional.

The following are the parent and child elements of the **PointerCap** element.

Parent elements
<a href="#">RadialPointer</a>

Child elements
<a href="#">PointerCap.CapImage</a>
<a href="#">PointerCap.CapStyle</a>
<a href="#">PointerCap.Hidden</a>
<a href="#">PointerCap.OnTop</a>
<a href="#">PointerCap.Reflection</a>
<a href="#">PointerCap.Style</a>
<a href="#">PointerCap.Width</a>

The following is the XML Schema definition of the **PointerCap** element in RDL 2008/01.

```
<xsd:complexType name="PointerCapType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="CapImage" type="CapImageType" minOccurs="0" />
    <xsd:element name="OnTop" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reflection" type="xsd:string" minOccurs="0" />
    <xsd:element name="CapStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **PointerCap** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="PointerCapType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="CapImage" type="CapImageType" minOccurs="0" />
    <xsd:element name="OnTop" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reflection" type="xsd:string" minOccurs="0" />
    <xsd:element name="CapStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

</xsd:complexType>

### 2.201.1 PointerCap.CapImage

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PointerCap.CapImage** element specifies the image to use for a [PointerCap](#) instance. The **PointerCap.CapImage** element is optional and is of type [CapImage](#).

The following is the parent element of the **PointerCap.CapImage** element.

Parent elements
PointerCap

The following is the XML Schema definition of the **PointerCap.CapImage** element.

```
<xsd:element name="CapImage" type="CapImageType" minOccurs="0">
```

### 2.201.2 PointerCap.CapStyle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **PointerCap.CapStyle** element specifies the type of a [PointerCap](#) instance. The **PointerCap.CapStyle** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**RoundedDark:** Specifies that the **PointerCap** instance is rounded and shaded darkly.

**Rounded:** Specifies that the **PointerCap** instance is rounded and shaded normally.

**RoundedLight:** Specifies that the **PointerCap** instance is rounded and shaded lightly.

**RoundedWithAdditionalTop:** Specifies that the **PointerCap** instance is rounded with an additional rounded top.

**RoundedWithWideIndentation:** Specifies that the **PointerCap** instance is rounded with a wide indentation in the middle.

**FlattenedWithIndentation:** Specifies that the **PointerCap** instance is round but flat with a regular-sized indentation in the middle.

**FlattenedWithWideIndentation:** Specifies that the **PointerCap** instance is round but flat with a wide indentation in the middle.

**RoundedGlossyWithIndentation:** Specifies that the **PointerCap** is rounded and glossy with a regular-sized indentation in the middle.

**RoundedWithIndentation:** Specifies that the **PointerCap** s rounded with a regular-sized indentation in the middle.

If the **PointerCap.CapStyle** element is not present, its value is interpreted as "RoundedDark".

The following is the parent element of the **PointerCap.CapStyle** element.

Parent elements
PointerCap

The following is the XML Schema definition of the **PointerCap.CapStyle** element.

```
<xsd:element name="CapStyle" type="xsd:string" minOccurs="0">
```

### 2.201.3 PointerCap.Hidden

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PointerCap.Hidden** element specifies whether the labels within a [RadialScale](#) or [LinearScale](#) element are hidden. The **PointerCap.Hidden** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **PointerCap.Hidden** element.

Parent elements
<a href="#">PointerCap</a>

The following is the XML Schema definition of the **PointerCap.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0">
```

### 2.201.4 PointerCap.OnTop

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PointerCap.OnTop** element specifies whether a [PointerCap](#) instance is displayed on top of a [RadialPointer](#). The **PointerCap.OnTop** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **PointerCap.OnTop** element.

Parent elements
PointerCap

The following is the XML Schema definition of the **PointerCap.OnTop** element.

```
<xsd:element name="OnTop" type="xsd:string" minOccurs="0">
```

### 2.201.5 PointerCap.Reflection

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PointerCap.Reflection** element specifies whether a [PointerCap](#) instance has a reflection effect. The **PointerCap.Reflection** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **PointerCap.Reflection** element.

Parent elements
PointerCap

The following is the XML Schema definition of the **PointerCap.Reflection** element.

```
<xsd:element name="Reflection" type="xsd:string" minOccurs="0">
```

### 2.201.6 PointerCap.Style

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PointerCap.Style** element specifies style properties for a [PointerCap](#) instance. The **PointerCap.Style** element is optional. This element is of type [Style](#).

The following is the parent element of the **PointerCap.Style** element.

Parent elements
PointerCap

The following is the XML Schema definition of the **PointerCap.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0">
```

### 2.201.7 PointerCap.Width

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PointerCap.Width** element specifies the width of a [PointerCap](#) instance as a percentage of the radius of its ancestor [RadialScale](#) instance. The **PointerCap.Width** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST greater than or equal to 0. [<55>](#) If this element is not present, its value is interpreted as 26.

The following is the parent element of the **PointerCap.Width** element.

Parent elements
PointerCap

The following is the XML Schema definition of the **PointerCap.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0">
```

## 2.202 CapImage

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CapImage** element specifies an image to use for a [PointerCap](#) instance within a [RadialGauge](#).

The following are the parent and child elements of the **CapImage** element.

<b>Parent elements</b>
PointerCap

<b>Child elements</b>
<a href="#">CapImage.MIMETYPE</a>
<a href="#">CapImage.Source</a>
<a href="#">CapImage.TransparentColor</a>
<a href="#">CapImage.Value</a>
<a href="#">CapImage.HueColor</a>
<a href="#">CapImage.OffsetX</a>
<a href="#">CapImage.OffsetY</a>

The following is the XML Schema definition of the **CapImage** element in RDL 2008/01.

```
<xsd:complexType name="CapImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="SizeType" minOccurs="0" />
    <xsd:element name="OffsetY" type="SizeType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **CapImage** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="CapImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="SizeType" minOccurs="0" />
    <xsd:element name="OffsetY" type="SizeType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.202.1 CapImage.MIMETYPE

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CapImage.MIMETYPE** element specifies the image format of a [CapImage](#) instance. The **CapImage.MIMETYPE** element is optional. If this element is present, its value MUST be a [ReportMIMETYPE](#).

If the peer [CapImage.Source](#) element is set to a value other than "Database", the **CapImage.MIMETYPE** element is ignored.

The following is the parent element of the **CapImage.MIMETYPE** element.

Parent elements
CapImage

The following is the XML Schema definition of the **CapImage.MIMETYPE** element.

```
<xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
```

## 2.202.2 CapImage.Source

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CapImage.Source** element specifies the type of the source that is associated with a [CapImage](#) instance. This element MUST be specified. The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**External:** Specifies that the peer [CapImage.Value](#) element contains a **String** constant or expression that evaluates to the location of an image.

**Embedded:** Specifies that the peer **CapImage.Value** element contains a **String** constant or expression that evaluates to the name of an [EmbeddedImage](#) within the report.

**Database:** Specifies that the peer **CapImage.Value** element contains an expression (such as a field in the database) that evaluates to the binary data for an image.

The following is the parent element of the **CapImage.Source** element.

Parent elements
CapImage

The following is the XML Schema definition of the **CapImage.Source** element.

```
<xsd:element name="Source" type="xsd:string" minOccurs="1" />
```

## 2.202.3 CapImage.TransparentColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CapImage.TransparentColor** element specifies the color to be treated as transparent in a [CapImage](#). This element is optional. If present, the value of the **CapImage.TransparentColor** element MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **CapImage.TransparentColor** element.

Parent elements
<a href="#">CapImage</a>

The following is the XML Schema definition of the **CapImage.TransparentColor** element.

```
<xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
```

## 2.202.4 CapImage.Value

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CapImage.Value** element specifies either the location or the actual data of an image, depending on the value of the peer [CapImage.Source](#) element. The **CapImage.Value** element MUST be specified.

If the peer **CapImage.Source** element is set to "External" and if the value of the **CapImage.Value** element is not empty, then the value of the **CapImage.Value** element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) constant or an expression that evaluates to the location of an image. This location MUST be a [ReportPath](#) or an [RdlURL](#) value.

If the peer **CapImage.Source** element is set to "Embedded" and if the value of the **CapImage.Value** element is not empty, then the value of the **CapImage.Value** element MUST be a **String** constant or an expression that evaluates to the name of a **CapImage.Value** in the report.

If the peer **CapImage.Source** element is set to "Database" and if the value of the **CapImage.Value** element is not empty, then the value of the **CapImage.Value** element MUST be an expression that evaluates to the binary data for an image.

If the **CapImage.Value** element has an empty value, the image MUST NOT be displayed.

The following is the parent element of the **CapImage.Value** element.

Parent elements
<a href="#">CapImage</a>

The following is the XML Schema definition of the **CapImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1" />
```

## 2.202.5 CapImage.HueColor

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CapImage.HueColor** element specifies the color with which to tint a [CapImage](#). The **CapImage.HueColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **CapImage.HueColor** element.

Parent elements
<a href="#">CapImage</a>

The following is the XML Schema definition of the **CapImage.HueColor** element.

```
<xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
```

## 2.202.6 CapImage.OffsetX

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CapImage.OffsetX** element specifies the X, or horizontal, offset for a [CapImage](#) instance within a [PointerCap](#) instance. The **CapImage.OffsetX** element is optional. If this element is present, its value MUST be an [RdlSize](#).

The following is the parent element of the **CapImage.OffsetX** element.

Parent elements
<a href="#">CapImage</a>

The following is the XML Schema definition of the **CapImage.OffsetX** element.

```
<xsd:element name="OffsetX" type="SizeType" minOccurs="0" />
```

## 2.202.7 CapImage.OffsetY

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CapImage.OffsetY** element specifies the Y, or vertical, offset for a [PointerCap.CapImage](#) instance. The **CapImage.OffsetY** element is optional. If this element is present, its value MUST be an [RdlSize](#).

The following is the parent element of the **CapImage.OffsetY** element.

Parent elements
<a href="#">CapImage</a>

The following is the XML Schema definition of the **CapImage.OffsetY** element.

```
<xsd:element name="OffsetY" type="SizeType" minOccurs="0" />
```

## 2.203 StateIndicators

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicators** element is ignored in any schema version prior to RDL 2010/01.

The following are the parent and child elements of the **StateIndicators** element.

Parent elements
<a href="#">GaugePanel</a>



<b>Child elements</b>
-----------------------

<a href="#">StateIndicators.StateIndicator</a>
--

The following is the XML Schema definition of the **StateIndicators** element in RDL 2008/01.

```
<xsd:complexType name="StateIndicatorsType">
  <xsd:sequence>
    <xsd:element name="StateIndicator" type="StateIndicatorType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **StateIndicators** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="StateIndicatorsType">
  <xsd:sequence>
    <xsd:element name="StateIndicator" type="StateIndicatorType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.203.1 StateIndicators.StateIndicator

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicators.StateIndicator** element is ignored in any schema version prior to RDL 2010/01. This element is of type [StateIndicator](#).

The following are the parent elements, attributes, and child elements of the **StateIndicator** element.

<b>Parent elements</b>
------------------------

<a href="#">StateIndicators</a>
---------------------------------

The following is the XML Schema definition of the **StateIndicators.StateIndicator** element.

```
<xsd:element name="StateIndicator" type="StateIndicatorType" minOccurs="1"
  maxOccurs="unbounded" />
```

### 2.204 StateIndicator

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicator** element is ignored in any schema version prior to RDL 2010/01.

The following are the parent elements, attributes, and child elements of the **StateIndicator** element.

<b>Parent elements</b>
------------------------

<a href="#">StateIndicators</a>
---------------------------------

Attributes
<a href="#">StateIndicator.Name</a>

Child elements
<a href="#">StateIndicator.ActionInfo</a>
<a href="#">StateIndicator.Height</a>
<a href="#">StateIndicator.Hidden</a>
<a href="#">StateIndicator.Left</a>
<a href="#">StateIndicator.MaximumValue</a>
<a href="#">StateIndicator.MinimumValue</a>
<a href="#">StateIndicator.ParentItem</a>
<a href="#">StateIndicator.ScaleFactor</a>
<a href="#">StateIndicator.StateDataElementName</a>
<a href="#">StateIndicator.StateDataElementOutput</a>
<a href="#">StateIndicator.ToolTip</a>
<a href="#">StateIndicator.Top</a>
<a href="#">StateIndicator.TransformationScope</a>
<a href="#">StateIndicator.TransformationType</a>
<a href="#">StateIndicator.Width</a>
<a href="#">StateIndicator.ZIndex</a>
<a href="#">StateIndicator.Angle</a>
<a href="#">StateIndicator.GaugeInputValue</a>
<a href="#">StateIndicator.IndicatorStates</a>
<a href="#">StateIndicator.IndicatorStyle</a>
<a href="#">StateIndicator.ResizeMode</a>
<a href="#">StateIndicator.StateImage</a>
<a href="#">StateIndicator.IndicatorImage</a>
<a href="#">StateIndicator.Style</a>
<a href="#">StateIndicator.Text</a>
<a href="#">StateIndicator.UseFontPercent</a>

The following is the XML Schema definition of the **StateIndicator** element in RDL 2008/01.

```
<xsd:complexType name="StateIndicatorType">
```

```

<xsd:choice minOccurs="0" maxOccurs="unbounded">
  <!--GaugePanelItemTypeStart-->
  <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
  <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
  <xsd:element name="Top" type="xsd:string" minOccurs="0" />
  <xsd:element name="Left" type="xsd:string" minOccurs="0" />
  <xsd:element name="Height" type="xsd:string" minOccurs="0" />
  <xsd:element name="Width" type="xsd:string" minOccurs="0" />
  <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
  <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
  <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
  <!--GaugePanelItemTypeEnd-->
  <xsd:element name="Style" type="StyleType" minOccurs="0" />
  <xsd:element name="GaugeInputValue" type="GaugeInputValueType"
    minOccurs="1" />
  <xsd:element name="IndicatorStates" type="IndicatorStatesType"
    minOccurs="0" />
  <xsd:element name="StateImage" type="StateImageType" minOccurs="0" />
  <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
  <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
  <xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="0" />
  <xsd:element name="Text" type="xsd:string" minOccurs="0" />
  <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />

  <xsd:any namespace="##other" processContents="skip" /> </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **StateIndicator** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="StateIndicatorType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType"
      minOccurs="0" />
    <xsd:element name="IndicatorStates" type="IndicatorStatesType"
      minOccurs="0" />
    <xsd:element name="IndicatorImage" type="IndicatorImageType" minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
    <xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransformationType" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransformationScope" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="ScaleFactor" type="xsd:string" minOccurs="0" />
    <xsd:element name="StateDataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="StateDataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:element>
    </xsd:choice>
  </xsd:complexType>

```

```

        </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.204.1 StateIndicator.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **StateIndicator.Name** attribute specifies a unique identifier for a [StateIndicator](#). This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier [UTR15]** within a [StateIndicators](#) collection.

The following is the parent element of the **StateIndicator.Name** attribute.

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.204.2 StateIndicator.ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **StateIndicator.ActionInfo** element specifies the actions for a [StateIndicator](#) element. The **StateIndicator.ActionInfo** element is optional. This element is ignored in any schema version prior to RDL 2010/01. This element is of type [ActionInfo](#).

The following is the parent element of the **StateIndicator.ActionInfo** element.

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.204.3 StateIndicator.Height

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **StateIndicator.Height** element specifies the height for a [StateIndicator](#) as a percentage of [StateIndicator.ParentItem](#). If **StateIndicator.ParentItem** is not specified, the value of the **StateIndicator.Height** element is interpreted as relative to the height of the [GaugePanel](#).

The **StateIndicator.Height** element is optional. This element is ignored if it is present in any schema version prior to RDL 2010/01. However, its data type is validated, and the value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If the **StateIndicator.Height** element is not present, its value is interpreted as 0.

The following is the parent element of the **StateIndicator.Height** element.

Parent elements
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **StateIndicator.Height** element.

```
<xsd:element name="Height" type="xsd:string" minOccurs="0" />
```

## 2.204.4 StateIndicator.Hidden

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicator.Hidden** element specifies whether a [StateIndicator](#) is hidden. The **StateIndicator.Hidden** element is optional. This element is ignored if it is present in any schema version prior to RDL 2010/01. However, its data type is validated, and the value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If the **StateIndicator.Hidden** element is not present, its value is interpreted as false.

The following is the parent element of the **StateIndicator.Hidden** element.

Parent elements
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **StateIndicator.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

## 2.204.5 StateIndicator.Left

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicator.Left** element specifies the distance from the left as a percentage of the [StateIndicator.ParentItem](#). If **StateIndicator.ParentItem** is not specified, the value of the **StateIndicator.Left** element is interpreted relative to the left of the [GaugePanel](#).

The **StateIndicator.Left** element is optional. This element is ignored if it is present in any schema version prior to RDL 2010/01. However, its data type is validated, and the value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If the [GaugePanel.AutoLayout](#) property for the **StateIndicator.Left** element's parent **GaugePanel** element is set to true, the **StateIndicator.Left** element is ignored. If the **StateIndicator.Left** element is not present, its value is interpreted as 0.

The following is the parent element of the **StateIndicator.Left** element.

Parent elements
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **StateIndicator.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

## 2.204.6 StateIndicator.MaximumValue

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **StateIndicator.MaximumValue** element specifies the maximum value used to transform the [StateIndicator.GaugeInputValue](#) element if the [StateIndicator.TransformationType](#) element is "Percentage". The **StateIndicator.MaximumValue** element is optional. This element is of type [GaugeInputValue](#).

The following is the parent element of the **StateIndicator.MaximumValue** element.

Parent elements
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **StateIndicator.MaximumValue** element.

```
<xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0">
```

## 2.204.7 StateIndicator.MinimumValue

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **StateIndicator.MinimumValue** element specifies the minimum value used to transform the **StateIndicator.GaugeInputValue.Value** if the [StateIndicator.TransformationType](#) is "Percentage". The **StateIndicator.MinimumValue** element is optional. This element is of type [GaugeInputValue](#).

The following is the parent element of the **StateIndicator.MinimumValue** element.

Parent elements
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **StateIndicator.MinimumValue** element.

```
<xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0">
```

## 2.204.8 StateIndicator.ParentItem

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicator.ParentItem** element specifies the name of the [StateIndicator](#) parent element. The **StateIndicator.ParentItem** element is ignored if it is present in any schema version prior to RDL 2010/01. However, its data type is validated, and the value MUST be a [String](#) ([XMLSCHEMA2/21](#) section 3.2.1) or an expression that evaluates to a **String**. The **StateIndicator.ParentItem** element is optional.

The following is the parent element of the **StateIndicator.ParentItem** element.

<b>Parent elements</b>
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **StateIndicator.ParentItem** element.

```
<xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
```

## 2.204.9 StateIndicator.ScaleFactor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicator.ScaleFactor** element specifies the scale factor to be applied to the state indicator. This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 1.

The following is the parent element of the **StateIndicator.ScaleFactor** element.

<b>Parent elements</b>
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **StateIndicator.ScaleFactor** element.

```
<xsd:element name="ScaleFactor" type="xsd:string" minOccurs="0" />
```

## 2.204.10 StateIndicator.StateDataElementName

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **StateIndicator.StateDataElementName** element specifies the name to use for the data element or attribute in a **data rendering** for the indicator state name that the state indicator value falls in its range. The **StateIndicator.StateDataElementName** element is optional. This element MUST be a **CLS-compliant identifier** [\[UTR15\]](#). The value of this element is interpreted as a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

The following is the parent element of the **StateIndicator.StateDataElementName** element.

<b>Parent elements</b>
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **StateIndicator.StateDataElementName** element.

```
<xsd:element name="StateDataElementName" type="xsd:string" minOccurs="0">
```

## 2.204.11 StateIndicator.StateDataElementOutput

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **StateIndicator.StateDataElementOutput** element indicates whether the indicator state name that the state indicator value falls in its range appears in a **data rendering**. The

**StateIndicator.StateDataElementOutput** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) constant that is one of the following:

**Output:** Specifies that the item appears in the data rendering output.

**NoOutput:** Specifies that the item does not appear in the data rendering output.

If this element is not present, its value is interpreted as "Output".

The following is the parent element of the **StateIndicator.StateDataElementOutput** element.

Parent elements
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **StateIndicator.StateDataElementOutput** element.

```
<xsd:element name="StateDataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.204.12 StateIndicator.ToolTip

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicator.ToolTip** element specifies the tooltip text for a [StateIndicator](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **StateIndicator.ToolTip** element is optional. This element is ignored if it is present in any schema version prior to RDL 2010/01. However, its data type is validated, and the value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **StateIndicator.ToolTip** element.

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

## 2.204.13 StateIndicator.Top

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicator.Top** element specifies the distance from the top as a percentage of [StateIndicator.ParentItem](#). If **StateIndicator.ParentItem** is not specified, the value of the **StateIndicator.Top** element is interpreted as relative to the top of the [GaugePanel](#).

The **StateIndicator.Top** element is optional. This element is ignored if it is present in any schema version prior to RDL 2010/01. However, its data type is validated, and the value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If the



[GaugePanel.AutoLayout](#) property for the **StateIndicator.Top** element's parent **GaugePanel** element is set to true, the **StateIndicator.Top** element is ignored.

If the **StateIndicator.Top** element is not present, its value is interpreted as 0. The following is the parent element of the **StateIndicator.Top** element.

Parent elements
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **StateIndicator.Top** element.

```
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
```

## 2.204.14 StateIndicator.TransformationScope

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **StateIndicator.TransformationScope** element specifies the scope to be used to calculate the **StateIndicator.MinimumValue.Value** and **StateIndicator.MaximumValue.Value** in case they are set to "NaN". The **StateIndicator.TransformationScope** element is optional. This element **MUST** be specified if the [StateIndicator.TransformationType](#) value is "Percentage" and either **StateIndicator.MinimumValue.Value** or **StateIndicator.MaximumValue.Value** is set to "NaN"; otherwise the **StateIndicator.TransformationScope** element is ignored.

The following is the parent element of the **StateIndicator.TransformationScope** element.

Parent elements
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **StateIndicator.TransformationScope** element.

```
<xsd:element name="TransformationScope" type="xsd:string" minOccurs="0" />
```

## 2.204.15 StateIndicator.TransformationType

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **StateIndicator.TransformationType** element specifies the transformation to be applied to the **StateIndicator.GaugeInputValue.Value** in a state indicator. This element is optional. If the **StateIndicator.TransformationType** element is present, its value **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element **MUST** be one of the following:

**None:** Specifies that no transformation is applied.

**Percentage:** Specifies that a percentage transformation is used.

If the **StateIndicator.TransformationType** element is not present, its value is interpreted as "None".

The following is the parent element of the **StateIndicator.TransformationType** element.

<b>Parent elements</b>
------------------------

<a href="#">StateIndicator</a>
--------------------------------

The following is the XML Schema definition of the **StateIndicator.TransformationType** element.

```
<xsd:element name="TransformationType" type="xsd:string" minOccurs="0" />
```

## 2.204.16 StateIndicator.Width

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicator.Width** element specifies the width of a [StateIndicator](#) as a percentage of [StateIndicator.ParentItem](#). If **StateIndicator.ParentItem** is not specified, the value of the **StateIndicator.Width** element is interpreted as relative to the width of the [GaugePanel](#).

The **StateIndicator.Width** element is optional. This is ignored if it is present in any schema version prior to RDL 2010/01. However, its data type is validated, and the value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If the **StateIndicator.Width** element is not present, its value is interpreted as 0.

The following is the parent element of the **StateIndicator.Width** element.

<b>Parent elements</b>
------------------------

<a href="#">StateIndicator</a>
--------------------------------

The following is the XML Schema definition of the **StateIndicator.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
```

## 2.204.17 StateIndicator.ZIndex

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicator.ZIndex** element specifies the drawing order of a [StateIndicator](#) within a [GaugePanel](#). The **StateIndicator.ZIndex** element is optional. This element is ignored if it is present in any schema version prior to RDL 2010/01. However, its data type is validated, and the value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 0. The value of this element MUST greater than or equal to 0 and less than or equal to 2147483647.

The following is the parent element of the **StateIndicator.ZIndex** element.

<b>Parent elements</b>
------------------------

<a href="#">StateIndicator</a>
--------------------------------

The following is the XML Schema definition of the **StateIndicator.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
```

## 2.204.18 StateIndicator.Angle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicator.Angle** element specifies the rotation angle of a [StateIndicator](#). The **StateIndicator.Angle** element is optional. This is ignored if it is present in any schema version prior to RDL 2010/01. However, its data type is validated, and the value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If **StateIndicator.Angle** is not specified, its value is interpreted as 0.

The following is the parent element of the **StateIndicator.Angle** element.

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.Angle** element.

```
<xsd:element name="Angle" type="xsd:string" minOccurs="0" />
```

## 2.204.19 StateIndicator.GaugeInputValue

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicator.GaugeInputValue** element specifies the value to use for a [StateIndicator](#) is ignored in any schema version prior to RDL 2010/01. This element is of type [GaugeInputValue](#).

The following is the parent element of the **StateIndicator.GaugeInputValue** element.

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.GaugeInputValue** element in RDL 2008/01.

```
<xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="1" />
```

The following is the XML Schema definition of the **StateIndicator.GaugeInputValue** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="0" />
```

## 2.204.20 StateIndicator.IndicatorStates

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicator.IndicatorStates** element specifies the set of indicator states within a [StateIndicator](#). The **StateIndicator.IndicatorStates** element is ignored in any schema version prior to RDL 2010/01. This element is of type [IndicatorStates](#).

The following is the parent element of the **StateIndicator.IndicatorStates** element.

<b>Parent elements</b>
StateIndicator

The following is the XML Schema definition of the **StateIndicator.IndicatorStates** element.

```
<xsd:element name="IndicatorStates" type="IndicatorStatesType" minOccurs="0" />
```

## 2.204.21 StateIndicator.IndicatorStyle

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicator.IndicatorStyle** element specifies the style of the state indicator if its value does not fall in any of the indicator states ranges. This element is ignored in any schema version prior to RDL 2010/01. This element is optional. If the **StateIndicator.IndicatorStyle** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following:

**None**: No marker specified.

**Image**: Specifies an image-based marker.

**Circle**: Specifies a circle style marker.

**Flag**: Specifies a flag style marker.

**ArrowDown**: Specifies an arrow down style marker.

**ArrowDownIncline**: Specifies an arrow down inclined style marker.

**ArrowSide**: Specifies an arrow side style marker.

**ArrowUp**: Specifies an arrow up style marker.

**ArrowUpIncline**: Specifies an arrow up inclined style marker.

**BoxesAllFilled**: Specifies a boxes all filled style marker.

**BoxesNoneFilled**: Specifies a boxes none filled style marker.

**BoxesOneFilled**: Specifies a boxes one filled style marker.

**BoxesTwoFilled**: Specifies a boxes two filled style marker.

**BoxesThreeFilled**: Specifies a boxes three filled style marker.

**QuartersAllFilled**: Specifies a quarters all filled style marker.

**QuartersNoneFilled**: Specifies a quarters none filled marker.

**QuartersOneFilled**: Specifies a quarters one filled style marker.

**QuartersTwoFilled**: Specifies a quarters two filled style marker.

**QuartersThreeFilled**: Specifies a quarters three filled style marker.

**SignalMeterFourFilled**: Specifies a signal meter four filled style marker.

**SignalMeterNoneFilled**: Specifies a signal meter none filled style marker.

**SignalMeterOneFilled**: Specifies a signal meter one fill style marker.

**SignalMeterThreeFilled**: Specifies a signal meter three filled style marker.

**SignalMeterTwoFilled**: Specifies a signal meter two filled style marker.

**StarQuartersAllFilled**: Specifies a star quarters all filled style marker.

**StarQuartersNoneFilled**: Specifies a star quarters none filled style marker.

**StarQuartersOneFilled**: Specifies a star quarters one filled style marker.

**StarQuartersTwoFilled**: Specifies a star quarters two filled style marker.

**StarQuartersThreeFilled**: Specifies a star quarters three filled style marker.

**ThreeSignsCircle**: Specifies a three signs circle style marker.

**ThreeSignsDiamond**: Specifies a three signs diamond style marker.

**ThreeSignsTriangle**: Specifies a three signs triangle style marker.

**ThreeSymbolCheck**: Specifies a three symbols check style marker.

**ThreeSymbolCross**: Specifies a three symbol cross style marker.

**ThreeSymbolExclamation**: Specifies a three symbol exclamation style marker.

**ThreeSymbolUnCircledCheck**: Specifies a three symbol uncircled check style marker.

**ThreeSymbolUnCircledCross**: Specifies a three symbol uncircled cross style marker.

**ThreeSymbolUnCircledExclamation**: Specifies a three symbol uncircled exclamation style marker.

**TrafficLight**: Specifies a traffic light style marker.

**TrafficLightUnrimmed**: Specifies a traffic light unrimmed style marker.

**TriangleDash**: Specifies a triangle dash style marker.

**TriangleDown**: Specifies a triangle down style marker.

**TriangleUp**: Specifies a triangle up style marker.

**ButtonStop**: Specifies a button stop style marker.

**ButtonPlay**: Specifies a button play style marker.

**ButtonPause**: Specifies a button pause style marker.

**FaceSmile**: Specifies a face smile style marker.

**FaceNeutral**: Specifies a face neutral style marker.

**FaceFrown**: Specifies a face frown style marker.

If the **StateIndicator.IndicatorStyle** element is not present, its value is interpreted as "Circle".

<b>Parent elements</b>
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **StateIndicator.IndicatorStyle** element.

```
<xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="0" />
```

## 2.204.22 StateIndicator.ResizeMode

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **StateIndicator.ResizeMode** element specifies whether content will resize to fit in the available space of a [StateIndicator](#). The **StateIndicator.ResizeMode** element is optional. This element is ignored in any schema version prior to RDL 2010/01. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. The specified **String** value for this element MUST be one of the following:

**AutoFit:** The content will automatically resize to fit in the available space of the state indicator.

**None:** The content will not automatically resize to fit in the available space of the state indicator.

If the **StateIndicator.ResizeMode** element is not present, its value is interpreted as "AutoFit". This element is ignored in any schema version prior to RDL 2010/01.

The following is the parent element of the **StateIndicator.ResizeMode** element.

Parent elements
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **StateIndicator.ResizeMode** element.

```
<xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
```

## 2.204.23 StateIndicator.IndicatorImage

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **StateIndicator.IndicatorImage** is of type [IndicatorImage](#).

The following is the parent element of the **StateIndicator.IndicatorImage** element.

Parent elements
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **StateIndicator.IndicatorImage** element.

```
<xsd:element name="IndicatorImage" type="IndicatorImageType" minOccurs="0" />
```

## 2.204.24 StateIndicator.StateImage

**Applies to** [RDL 2008/01](#)

The **StateIndicator.StateImage** element is ignored. This element is of type [StateImage](#).

The following is the parent element of the **StateIndicator.StateImage** element.

<b>Parent elements</b>
------------------------

<a href="#">StateIndicator</a>
--------------------------------

The following is the XML Schema definition of the **StateIndicator.StateImage** element.

```
<xsd:element name="StateImage" type="StateImageType" minOccurs="0" />
```

## 2.204.25 StateIndicator.Style

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **StateIndicator.Style** element specifies style information for a [StateIndicator](#). The **StateIndicator.Style** element is ignored in any schema version prior to RDL 2010/01. This element is optional. This element is of type [Style](#).

The following is the parent element of the **StateIndicator.Style** element.

<b>Parent elements</b>
------------------------

StateIndicator
----------------

The following is the XML Schema definition of the **StateIndicator.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.204.26 StateIndicator.Text

*Applies to [RDL 2008/01](#)*

The **StateIndicator.Text** element is ignored.

The following is the parent element of the **StateIndicator.Text** element.

<b>Parent elements</b>
------------------------

<a href="#">StateIndicator</a>
--------------------------------

The following is the XML Schema definition of the **StateIndicator.Text** element.

```
<xsd:element name="Text" type="xsd:string" minOccurs="0" />
```

## 2.204.27 StateIndicator.UseFontPercent

*Applies to [RDL 2008/01](#)*

The **StateIndicator.UseFontPercent** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **StateIndicator.UseFontPercent** element.

<b>Parent elements</b>
------------------------

<a href="#">StateIndicator</a>
--------------------------------

The following is the XML Schema definition of the **StateIndicator.UseFontPercent** element.

```
<xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
```

## 2.205 IndicatorStates

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **IndicatorStates** element is ignored in any schema version prior to RDL 2010/01.

The following are the parent and child elements of the **IndicatorStates** element.

<b>Parent elements</b>
------------------------

<a href="#">StateIndicator</a>
--------------------------------

<b>Child elements</b>
-----------------------

<a href="#">IndicatorStates.IndicatorState</a>
--

The following is the XML Schema definition of the **IndicatorStates** element in RDL 2008/01.

```
<xsd:complexType name="IndicatorStatesType">
  <xsd:sequence>
    <xsd:element name="IndicatorState" type="IndicatorStateType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **IndicatorStates** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="IndicatorStatesType">
  <xsd:sequence>
    <xsd:element name="IndicatorState" type="IndicatorStateType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.205.1 IndicatorStates.IndicatorState

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **IndicatorStates.IndicatorState** element is ignored in any schema version prior to RDL 2010/01. This element is of type [IndicatorState](#).

The following is the parent element of the **IndicatorStates.IndicatorState** element.



<b>Parent elements</b>
<a href="#">IndicatorStates</a>

The following is the XML Schema definition of the **IndicatorStates.IndicatorState** element.

```
<xsd:element name="IndicatorState" type="IndicatorStateType" minOccurs="1"
maxOccurs="unbounded" />
```

## 2.206 IndicatorState

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **IndicatorState** element is ignored in any schema version prior to RDL 2010/01.

The following are the parent elements, attributes, and child elements of the **IndicatorState** element.

<b>Parent elements</b>
<a href="#">IndicatorStates</a>

<b>Attributes</b>
<a href="#">IndicatorState.Name</a>

<b>Child elements</b>
<a href="#">IndicatorState.Color</a>
<a href="#">IndicatorState.IndicatorStyle</a>
<a href="#">IndicatorState.ScaleFactor</a>
<a href="#">IndicatorState.Style</a>
<a href="#">IndicatorState.StartValue</a>
<a href="#">IndicatorState.EndValue</a>
<a href="#">IndicatorState.StateImage</a>
<a href="#">IndicatorState.IndicatorImage</a>
<a href="#">IndicatorState.Text</a>

The following is the XML Schema definition of the **IndicatorState** element in RDL 2008/01.

```
<xsd:complexType name="IndicatorStateType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="StartValue" type="GaugeInputValue" minOccurs="0" />
    <xsd:element name="EndValue" type="GaugeInputValue" minOccurs="0" />
    <xsd:element name="StateImage" type="StateImageType" minOccurs="0" />
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
</xsd:complexType>
```

```

</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **IndicatorState** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="IndicatorStateType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="StartValue" type="GaugeInputValue" minOccurs="0" />
    <xsd:element name="EndValue" type="GaugeInputValue" minOccurs="0" />
    <xsd:element name="IndicatorImage" type="IndicatorImageType" minOccurs="0" />
    <xsd:element name="Color" type="xsd:string" minOccurs="1" />
    <xsd:element name="ScaleFactor" type="xsd:string" minOccurs="1" />
    <xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.206.1 IndicatorState.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **IndicatorState.Name** attribute specifies a unique identifier for an [IndicatorState](#). The **IndicatorState.Name** attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#).

The following is the parent element of the **IndicatorState.Name** attribute.

Parent elements
<a href="#">IndicatorState</a>

The following is the XML Schema definition of the **IndicatorState.Name** attribute.

```

<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />

```

### 2.206.2 IndicatorState.Color

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **IndicatorState.Color** element specifies the color of the state indicator if its value falls in the range of this indicator state element. The **IndicatorState.Color** element MUST be specified. The value of the **IndicatorState.Color** element MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

Following is the parent element of the **IndicatorState.Color** element.

Parent elements
<a href="#">IndicatorState</a>

The following is the XML Schema definition of the **IndicatorState.Color** element.

```
<xsd:element name="Color" type="xsd:string" minOccurs="1" />
```

### 2.206.3 IndicatorState.IndicatorStyle

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **IndicatorState.IndicatorStyle** specifies the style of the state indicator if its value falls in the indicator state ranges. This element **MUST** be specified, and its value **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element **MUST** be one of the following:

**None:** No marker is specified.

**Image:** Specifies an image-based marker.

**Circle:** Specifies a circle style marker.

**Flag:** Specifies a flag style marker.

**ArrowDown:** Specifies an arrow down style marker.

**ArrowDownIncline:** Specifies an arrow down inclined style marker.

**ArrowSide:** Specifies an arrow side style marker.

**ArrowUp:** Specifies an arrow up style marker.

**ArrowUpIncline:** Specifies an arrow up inclined style marker.

**BoxesAllFilled:** Specifies a boxes all filled style marker.

**BoxesNoneFilled:** Specifies a boxes none filled style marker.

**BoxesOneFilled:** Specifies a boxes one filled style marker.

**BoxesTwoFilled:** Specifies a boxes two filled style.

**BoxesThreeFilled:** Specifies a boxes three filled style marker.

**ButtonPause:** Specifies a button pause style marker.

**ButtonPlay:** Specifies a button play style marker.

**ButtonStop:** Specifies a button stop style marker.

**FaceSmile:** Specifies a face smile style marker.

**LightArrowDown:** Specifies a light arrow down style marker.

**LightArrowDownIncline:** Specifies a light arrow down inclined style marker.

**LightArrowSide:** Specifies a light arrow side style marker.

**LightArrowUp:** Specifies a light arrow up style marker.

**LightArrowUpIncline:** Specifies a light arrow up inclined style marker.

**QuartersAllFilled:** Specifies a quarters all filled style marker.

**QuartersNoneFilled:** Specifies a quarters none filled marker.

**QuartersOneFilled**: Specifies a quarters one filled style marker.

**QuartersTwoFilled**: Specifies a quarters two filled style marker.

**QuartersThreeFilled**: Specifies a quarters three filled style marker.

**SignalMeterNoneFilled**: Specifies a signal meter none filled style marker.

**SignalMeterOneFill**: Specifies a signal meter one fill style marker.

**SignalMeterTwoFilled**: Specifies a signal meter two filled style marker.

**SignalMeterThreeFilled**: Specifies a signal meter three filled style marker.

**SignalMeterFourFilled**: Specifies a signal meter four filled style marker.

**StarQuartersAllFilled**: Specifies a start quarters all filled style marker.

**StarQuartersNoneFilled**: Specifies a star quarters none filled style marker.

**StarQuartersOneFilled**: Specifies a star quarters one filled style marker.

**StarQuartersTwoFilled**: Specifies a star quarters two filled style marker.

**StarQuartersThreeFilled**: Specifies a star quarters three filled style marker.

**ThreeSignsCircle**: Specifies a three signs circle style marker.

**ThreeSignsDiamond**: Specifies a three signs diamond style marker.

**ThreeSignsTriangle**: Specifies a three signs triangle style marker.

**ThreeSymbolCheck**: Specifies a three symbols check style marker.

**ThreeSymbolCross**: Specifies a three symbol cross style marker.

**ThreeSymbolExclamation**: Specifies a three symbol exclamation style marker.

**ThreeSymbolUnCircledCheck**: Specifies a three symbol uncircled check style marker.

**ThreeSymbolUnCircledCross**: Specifies a three symbol uncircled cross style style marker.

**ThreeSymbolUnCircledExclamation**: Specifies a three symbol uncircled exclamation style marker.

**TrafficLight**: Specifies a traffic light style marker.

**TrafficLightUnrimmed**: Specifies a traffic light unrimmed style marker.

**TriangleDash**: Specifies a triangle dash style marker.

**TriangleDown**: Specifies a triangle down style marker.

**TriangleUp**: Specifies a triangle up style marker.

The following is the parent element of the **IndicatorState.IndicatorStyle** element.

Parent elements
<a href="#">IndicatorState</a>

The following is the XML Schema definition of the **IndicatorState.IndicatorStyle** element.

```
<xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="1" />
```

## 2.206.4 IndicatorState.Style

**Applies to** [RDL 2008/01](#)

The **IndicatorState.Style** element is ignored. This element is of type [Style](#).

The following is the parent element of the **IndicatorState.Style** element.

Parent elements
<a href="#">IndicatorState</a>

The following is the XML Schema definition of the **IndicatorState.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.206.5 IndicatorState.StartValue

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **IndicatorState.StartValue** element specifies the start value of an [IndicatorState](#). This element is ignored in any schema version prior to RDL 2010/01. This element is of type [GaugeInputValue](#).

The following is the parent element of the **IndicatorState.StartValue** element.

Parent elements
IndicatorState

The following is the XML Schema definition of the **IndicatorState.StartValue** element.

```
<xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
```

## 2.206.6 IndicatorState.EndValue

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **IndicatorState.EndValue** element specifies the end value for an [IndicatorState](#). This element is ignored in any schema version prior to RDL 2010/01. This element is of type [GaugeInputValue](#).

The following is the parent element of the **IndicatorState.EndValue** element.

Parent elements
IndicatorState

The following is the XML Schema definition of the **IndicatorState.EndValue** element.

```
<xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
```

## 2.206.7 IndicatorState.IndicatorImage

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **IndicatorState.IndicatorImage** is of type [IndicatorImage](#).

The following is the parent element of the **IndicatorState.IndicatorImage** element.

Parent elements
<a href="#">IndicatorState</a>

The following is the XML Schema definition of the **IndicatorState.IndicatorImage** element.

```
<xsd:element name="IndicatorImage" type="IndicatorImageType" minOccurs="0" />
```

## 2.206.8 IndicatorState.ScaleFactor

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **IndicatorState.ScaleFactor** element specifies the scale factor to be applied to the state indicator in case the state indicator value falls in the range of the indicator state. This element **MUST** be specified, and its value **MUST** be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**.

Following is the parent element of the **IndicatorState.ScaleFactor** element.

Parent elements
<a href="#">StateIndicator</a>

The following is the XML Schema definition of the **IndicatorState.ScaleFactor** element.

```
<xsd:element name="ScaleFactor" type="xsd:string" minOccurs="1" />
```

## 2.206.9 IndicatorState.StateImage

*Applies to [RDL 2008/01](#)*

The **IndicatorState.StateImage** element is ignored. This element is of type [StateImage](#).

The following is the parent element of the **IndicatorState.StateImage** element.

Parent elements
<a href="#">IndicatorState</a>

The following is the XML Schema definition of the **IndicatorState.StateImage** element.

```
<xsd:element name="StateImage" type="StateImageType" minOccurs="0" />
```

## 2.206.10 IndicatorState.Text

Applies to [RDL 2008/01](#)

The **IndicatorState.Text** element is ignored if it is present. However, its data type is validated, and the value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **IndicatorState.Text** element.

Parent elements
<a href="#">IndicatorState</a>

The following is the XML Schema definition of the **IndicatorState.Text** element.

```
<xsd:element name="Text" type="xsd:string" minOccurs="0" />
```

## 2.207 StateImage

Applies to [RDL 2008/01](#)

The **StateImage** element is ignored.

The following are the parent and child elements of the **StateImage** element.

Parent elements
<a href="#">StateIndicator</a>

Child elements
<a href="#">StateImage.MIMETYPE</a>
<a href="#">StateImage.Source</a>
<a href="#">StateImage.TransparentColor</a>
<a href="#">StateImage.Value</a>
<a href="#">StateImage.HueColor</a>
<a href="#">StateImage.Transparency</a>

The following is the XML Schema definition of the **StateImage** element.

```
<xsd:complexType name="StateImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
</xsd:complexType>
```

```
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

## 2.207.1 StateImage.MIMETYPE

**Applies to** [RDL 2008/01](#)

The **StateImage.MIMETYPE** element is ignored.

The following is the parent element of the **StateImage.MIMETYPE** element.

Parent elements
<a href="#">StateImage</a>

The following is the XML Schema definition of the **StateImage.MIMETYPE** element.

```
<xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0">
```

## 2.207.2 StateImage.Source

**Applies to** [RDL 2008/01](#)

The **StateImage.Source** element is ignored.

The following is the parent element of the **StateImage.Source** element.

Parent elements
<a href="#">StateImage</a>

The following is the XML Schema definition of the **StateImage.Source** element.

```
<xsd:element name="Source" type="xsd:string" minOccurs="1">
```

## 2.207.3 StateImage.TransparentColor

**Applies to** [RDL 2008/01](#)

The **StateImage.TransparentColor** element is ignored if it is present. However, its data type is validated, and the value **MUST** be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **StateImage.TransparentColor** element.

Parent elements
<a href="#">StateImage</a>

The following is the XML Schema definition of the **StateImage.TransparentColor** element.

```
<xsd:element name="TransparentColor" type="xsd:string" minOccurs="0">
```



## 2.207.4 StateImage.Value

*Applies to [RDL 2008/01](#)*

The **StateImage.Value** element is ignored.

The following is the parent element of the **StateImage.Value** element.

Parent elements
<a href="#">StateImage</a>

The following is the XML Schema definition of the **StateImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1">
```

## 2.207.5 StateImage.HueColor

*Applies to [RDL 2008/01](#)*

The **StateImage.HueColor** element is ignored if it is present. However, its data type is validated, and the value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **StateImage.HueColor** element.

Parent elements
<a href="#">StateImage</a>

The following is the XML Schema definition of the **StateImage.HueColor** element.

```
<xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
```

## 2.207.6 StateImage.Transparency

*Applies to [RDL 2008/01](#)*

The **StateImage.Transparency** element is ignored if it is present. However, its data type is validated, and the value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**.

Following is the parent element of the **StateImage.Transparency** element.

Parent elements
<a href="#">StateImage</a>

The following is the XML Schema definition of the **StateImage.Transparency** element.

```
<xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
```

## 2.208 IndicatorImage

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The following are the parent and child elements of the **IndicatorImage** element.

Parent elements
<a href="#">StateIndicator</a>
<a href="#">IndicatorState</a>

Child elements
<a href="#">IndicatorImage.MIMETYPE</a>
<a href="#">IndicatorImage.Source</a>
<a href="#">IndicatorImage.TransparentColor</a>
<a href="#">IndicatorImage.Value</a>
<a href="#">IndicatorImage.HueColor</a>
<a href="#">IndicatorImage.Transparency</a>

The following is the XML Schema definition of the **IndicatorImage** element.

```
<xsd:complexType name="IndicatorImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.208.1 IndicatorImage.MIMETYPE

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **IndicatorImage.MIMETYPE** element specifies the image format of an [IndicatorImage](#). The **IndicatorImage.MIMETYPE** element is optional. If this element is present, its value MUST be a [ReportMIMETYPE](#).

If the peer element [IndicatorImage.Source](#) is set to a value other than "Database", the **IndicatorImage.MIMETYPE** element is ignored.

The following is the parent element of the **IndicatorImage.MIMETYPE** element.

Parent elements
<a href="#">IndicatorImage</a>

The following is the XML Schema definition of the **IndicatorImage.MIMETYPE** element.

```
<xsd:element name="MIMEType" type="xsd:string" minOccurs="0">
```

## 2.208.2 IndicatorImage.Source

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **IndicatorImage.Source** element specifies the type of source associated with an [IndicatorImage](#). The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following:

**External:** Specifies that the peer [IndicatorImage.Value](#) element contains a **String** constant or expression that evaluates to the location of an image.

**Embedded:** Specifies that the peer **IndicatorImage.Value** element contains a **String** constant or expression that evaluates to the name of an [EmbeddedImage](#) within the report.

**Database:** Specifies that the peer **IndicatorImage.Value** element contains an expression (for example, a field in the database) that evaluates to the binary data for an image.

The **IndicatorImage.Source** element MUST be specified.

The following is the parent element of the **IndicatorImage.Source** element.

Parent elements
IndicatorImage

The following is the XML Schema definition of the **IndicatorImage.Source** element.

```
<xsd:element name="Source" type="xsd:string" minOccurs="1">
```

## 2.208.3 IndicatorImage.TransparentColor

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **IndicatorImage.TransparentColor** element specifies the color to treat as transparent in an [IndicatorImage](#). This element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **IndicatorImage.TransparentColor** element.

Parent elements
IndicatorImage

The following is the XML Schema definition of the **IndicatorImage.TransparentColor** element.

```
<xsd:element name="TransparentColor" type="xsd:string" minOccurs="0">
```

## 2.208.4 IndicatorImage.Value

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **IndicatorImage.Value** element specifies the location of an [IndicatorImage](#), depending on the peer [IndicatorImage.Source](#) element. The **IndicatorImage.Value** element MUST be specified.

If the peer **IndicatorImage.Source** element is set to "External" and if the value of **IndicatorImage.Value** is non-empty, then the value of **IndicatorImage.Value** MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) constant or an expression that evaluates to the location of an image. This location MUST be expressed as a [ReportPath](#) or [RdlColor](#).

If the peer **IndicatorImage.Source** element is set to "Embedded" and if the value of **IndicatorImage.Value** is non-empty, then the value of **IndicatorImage.Value** MUST be a **String** constant or an expression that evaluates to the name of an [EmbeddedImage](#) within the report.

If the peer **IndicatorImage.Source** element is set to "Database" and if its value is non-empty, its value MUST be an expression that evaluates to the binary data for an image. If the **IndicatorImage.Value** element has an empty value, the image MUST NOT be displayed.

The following is the parent element of the **IndicatorImage.Value** element.

Parent elements
IndicatorImage

The following is the XML Schema definition of the **IndicatorImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1">
```

## 2.208.5 IndicatorImage.HueColor

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **IndicatorImage.HueColor** element specifies the color to tint an [IndicatorImage](#). The **IndicatorImage.HueColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

The following is the parent element of the **IndicatorImage.HueColor** element.

Parent elements
IndicatorImage

The following is the XML Schema definition of the **IndicatorImage.HueColor** element.

```
<xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
```

## 2.208.6 IndicatorImage.Transparency

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **IndicatorImage.Transparency** element specifies the percentage of transparency for an [IndicatorImage](#). The **IndicatorImage.Transparency** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The value of this element MUST be greater than or equal to 0 and less than or equal to 100. If the **IndicatorImage.Transparency** element is not present, its value is interpreted as "0".

Following is the parent element of the **IndicatorImage.Transparency** element.

Parent elements
IndicatorImage

The following is the XML Schema definition of the **IndicatorImage.Transparency** element.

```
<xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
```

## 2.209 Map

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map** element specifies **map**-based data visualization.

The following are the parent elements, attributes, and child elements of the **Map** element.

Parent elements
<a href="#">ReportItems</a>
<a href="#">CellContents</a>
<a href="#">CustomReportItem.AltReportItem</a>

Attributes
<a href="#">Map.Name</a>

Child elements
<a href="#">Map.ActionInfo</a>
<a href="#">Map.AntiAliasing</a>
<a href="#">Map.Bookmark</a>
<a href="#">Map.CustomProperties</a>
<a href="#">Map.DataElementName</a>
<a href="#">Map.DataElementOutput</a>
<a href="#">Map.DocumentMapLabel</a>
<a href="#">Map.Height</a>
<a href="#">Map.Left</a>
<a href="#">Map.MapBorderSkin</a>
<a href="#">Map.MapColorScale</a>
<a href="#">Map.MapDataRegions</a>
<a href="#">Map.MapDistanceScale</a>

Child elements
<a href="#">Map.MapLayers</a>
<a href="#">Map.MapLegends</a>
<a href="#">Map.MapTitles</a>
<a href="#">Map.MapViewport</a>
<a href="#">Map.MaximumSpatialElementCount</a>
<a href="#">Map.MaximumTotalPointCount</a>
<a href="#">Map.PageBreak</a>
<a href="#">Map.PageName</a>
<a href="#">Map.RepeatWith</a>
<a href="#">Map.ShadowIntensity</a>
<a href="#">Map.Style</a>
<a href="#">Map.TextAntiAliasingQuality</a>
<a href="#">Map.TileLanguage</a>
<a href="#">Map.ToolTip</a>
<a href="#">Map.Top</a>
<a href="#">Map.Visibility</a>
<a href="#">Map.Width</a>
<a href="#">Map.ZIndex</a>

The following is the XML Schema definition of the **Map** element.

```

<xsd:complexType name="MapType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--ReportItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--ReportItemTypeEnd-->
  </xsd:choice>
</xsd:complexType>

```

```

<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
<xsd:element name="PageName" type="xsd:string" minOccurs="0" />
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="TileLanguage" type="xsd:string" minOccurs="0" />
<xsd:element name="MapLayers" type="MapLayersType" minOccurs="0" />
<xsd:element name="MapDataRegions" type="MapDataRegionsType" minOccurs="0" />
<xsd:element name="MapViewport" type="MapViewportType" minOccurs="1" />
<xsd:element name="MapLegends" type="MapLegendsType" minOccurs="0" />
<xsd:element name="MapTitles" type="MapTitlesType" minOccurs="0" />
<xsd:element name="MapDistanceScale" type="MapDistanceScaleType"
minOccurs="0" />
<xsd:element name="MapColorScale" type="MapColorScaleType" minOccurs="0" />
<xsd:element name="MapBorderSkin" type="MapBorderSkinType" minOccurs="0" />
<xsd:element name="AntiAliasing" type="xsd:string" minOccurs="0" />
<xsd:element name="TextAntiAliasingQuality" type="xsd:string" minOccurs="0" />
<xsd:element name="ShadowIntensity" type="xsd:string" minOccurs="0" />
<xsd:element name="MaximumSpatialElementCount" type="xsd:unsignedInt"
minOccurs="0" />
<xsd:element name="MaximumTotalPointCount" type="xsd:unsignedInt"
minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.209.1 Map.Name

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.Name** attribute specifies the name of a map. This attribute is required. The value of this attribute **MUST** be a case-sensitive [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is a **CLS-compliant identifier** [UTR15]. This value **MUST** be unique among **data regions** and groups in the [Report](#).

The following is the parent element of the **Map.Name** attribute.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.209.2 Map.ActionInfo

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.ActionInfo** element specifies a set of actions for a [Map](#) element. The **Map.ActionInfo** element is optional. This element is of type [ActionInfo](#).

The following is the parent element of the **Map.ActionInfo** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.209.3 Map.AntiAliasing

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **Map.AntiAliasing** element specifies the **anti-aliasing** type for a Map. This element is optional. If the **Map.AntiAliasing** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**All:** Both text and graphics have anti-aliasing applied.

**Text:** Only text has anti-aliasing applied.

**Graphics:** Only graphics have anti-aliasing applied.

**None:** Neither text or graphics have anti-aliasing applied.

If this element is not present, its value is interpreted as "All".

The following is the parent element of the **Map.AntiAliasing** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.AntiAliasing** element.

```
<xsd:element name="AntiAliasing" type="xsd:string" minOccurs="0" />
```

### 2.209.4 Map.Bookmark

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **Map.Bookmark** element specifies a **bookmark** for a Map that can be linked to via an action with [Action.BookmarkLink](#) set. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **Map.Bookmark** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.Bookmark** element.

```
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
```

### 2.209.5 Map.CustomProperties

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*



The **Map.CustomProperties** element specifies custom information for a [Map](#) that will be handed to a report rendering component. The **Map.CustomProperties** element is optional. This element is of type [CustomProperties](#).

The following is the parent element of the **Map.CustomProperties** element.

Parent elements
Map

The following is the XML Schema definition of the **Map.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

## 2.209.6 Map.DataElementName

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.DataElementName** element specifies the name to use for the **data element** or attribute of a [Map](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as the **Name** attribute of the map.

The following is the parent element of the **Map.DataElementName** element.

Parent elements
Map

The following is the XML Schema definition of the **Map.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

## 2.209.7 Map.DataElementOutput

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.DataElementOutput** element specifies whether a [Map](#) appears in a data (XML, CSV) rendering. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**Output:** Specifies that the map instance appears in a **data rendering** output.

**NoOutput:** Specifies that the map instance does not appear in a data rendering output.

The following is the parent element of the **Map.DataElementOutput** element.

Parent elements
Map

The following is the XML Schema definition of the **Map.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">  
  <xsd:simpleType>  
    <xsd:restriction base="xsd:string">
```

```

    <xsd:enumeration value="Output" />
    <xsd:enumeration value="NoOutput" />
  </xsd:restriction>
</xsd:simpleType>
</xsd:element>

```

## 2.209.8 Map.DocumentMapLabel

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.DocumentMapLabel** element specifies a label to identify a Map instance within the client UI in order to provide a user-friendly label for searching. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **Map.DocumentMapLabel** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.DocumentMapLabel** element.

```
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
```

## 2.209.9 Map.Height

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.Height** element specifies the height of a [Map](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as the height of the map's container minus the value of the peer [Map.Top](#) element, if specified.

The following is the parent element of the **Map.Height** element.

Parent elements
Map

The following is the XML Schema definition of the **Map.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="0" />
```

## 2.209.10 Map.Left

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.Left** element specifies the distance of a map from the left of the height of the map's container. This element is optional. If this element is present, its value MUST be an [RdlSize](#). If the **Map.Left** element is not present, its value is interpreted as 0.

The following is the parent element of the **Map.Left** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.Left** element.

```
<xsd:element name="Left" type="SizeType" minOccurs="0" />
```

### 2.209.11 Map.MapBorderSkin

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **Map.MapBorderSkin** element specifies a **border skin** for a map. This element is optional. This element is of type [MapBorderSkin](#).

The following is the parent element of the **Map.MapBorderSkin** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.MapBorderSkin** element.

```
<xsd:element name="MapBorderSkin" type="MapBorderSkinType" minOccurs="0" />
```

### 2.209.12 Map.MapColorScale

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **Map.MapColorScale** element specifies the color scale for a [Map](#). This element is optional. This element is of type [MapColorScale](#).

The following is the parent element of the **Map.MapColorScale** element.

Parent elements
Map

The following is the XML Schema definition of the **Map.MapColorScale** element.

```
<xsd:element name="MapColorScale" type="MapColorScaleType" minOccurs="0" />
```

### 2.209.13 Map.MapDataRegions

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **Map.MapDataRegions** element specifies a collection of [MapDataRegion](#) elements for a Map. The **Map.MapDataRegions** element is optional. This element is of type [MapDataRegions](#).

The following is the parent element of the **Map.MapDataRegions** element.

<b>Parent elements</b>
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.MapDataRegions** element.

```
<xsd:element name="MapDataRegions" type="MapDataRegionsType" minOccurs="0" />
```

## 2.209.14 Map.MapDistanceScale

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.MapDistanceScale** element specifies the distance scale for a Map. This element is optional. This element is of type [MapDistanceScale](#).

The following is the parent element of the **Map.MapDistanceScale** element.

<b>Parent elements</b>
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.MapDistanceScale** element.

```
<xsd:element name="MapDistanceScale" type="MapDistanceScaleType" minOccurs="0" />
```

## 2.209.15 Map.MapLayers

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.MapLayers** element specifies a collection of layers for a Map. This element is optional. This element is of type [MapLayers](#).

The following is the parent element of the **Map.MapLayers** element.

<b>Parent elements</b>
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.MapLayers** element.

```
<xsd:element name="MapLayers" type="MapLayersType" minOccurs="0" />
```

## 2.209.16 Map.MapLegends

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.MapLegends** element specifies a collection of map legends (specified by [MapLegend](#) elements) to be drawn in a [Map](#). This element is optional. This element is of type [MapLegends](#).

The following is the parent element of the **Map.MapLegends** element.

Parent elements
Map

The following is the XML Schema definition of the **Map.MapLegends** element.

```
<xsd:element name="MapLegends" type="MapLegendsType" minOccurs="0" />
```

## 2.209.17 Map.MapTitles

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.MapTitles** element specifies a collection of map titles to be drawn in a Map. This element is optional. This element is of type [MapTitles](#).

The following is the parent element of the **Map.MapTitles** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.MapTitles** element.

```
<xsd:element name="MapTitles" type="MapTitlesType" minOccurs="0" />
```

## 2.209.18 Map.MapViewport

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.MapViewport** element specifies the [MapViewport](#) for a Map. This element MUST be specified. This element is of type **MapViewport**.

The following is the parent element of the **Map.MapViewport** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.MapViewport** element.

```
<xsd:element name="MapViewport" type="MapViewportType" minOccurs="1" />
```

## 2.209.19 Map.MaximumSpatialElementCount

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.MaximumSpatialElementCount** element specifies the maximum number of **spatial elements** that are allowed in a [Map](#). This element is optional. If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 20000.

The following is the parent element of the **Map.MaximumSpatialElementCount** element.

Parent elements
Map

The following is the XML Schema definition of the **Map.MaximumSpatialElementCount** element.

```
<xsd:element name="MaximumSpatialElementCount" type="xsd:unsignedInt" minOccurs="0" />
```

## 2.209.20 Map.MaximumTotalPointCount

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.MaximumTotalPointCount** element specifies the maximum total number of **map points** in all **spatial elements** that are allowed in a [Map](#). The **Map.MaximumTotalPointCount** element is optional.

If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 1000000.

The following is the parent element of the **Map.MaximumTotalPointCount** element.

Parent elements
Map

The following is the XML Schema definition of the **Map.MaximumTotalPointCount** element.

```
<xsd:element name="MaximumTotalPointCount" type="xsd:unsignedInt" minOccurs="0" />
```

## 2.209.21 Map.PageBreak

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.PageBreak** element specifies page break behavior for a Map. This element is optional. This element is of type [PageBreak](#).

The following is the parent element of the **Map.PageBreak** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.PageBreak** element.

```
<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
```

## 2.209.22 Map.PageName

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.PageName** element specifies value to use for the name of a paginated page. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Map.PageName** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.PageName** element.

```
<xsd:element name="PageName" type="xsd:string" minOccurs="0" />
```

### 2.209.23 **Map.RepeatWith**

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **Map.RepeatWith** element specifies the name of a data region in which a [Map](#) is repeated if that data region spans multiple pages. The data region MUST be in the same [ReportItems](#) collection as this map instance. If this map instance is within a [PageSection](#), the **Map.RepeatWith** element is ignored.

This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

The following is the parent element of the **Map.RepeatWith** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.RepeatWith** element.

```
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
```

### 2.209.24 **Map.ShadowIntensity**

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **Map.ShadowIntensity** element specifies the intensity of the shadows throughout a Map. This element is optional. If the **Map.ShadowIntensity** element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 25.

The following is the parent element of the **Map.ShadowIntensity** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.ShadowIntensity** element.

```
<xsd:element name="ShadowIntensity" type="xsd:string" minOccurs="0" />
```

### 2.209.25 **Map.Style**

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **Map.Style** element specifies style information for a Map. This element is optional. This element is of type [Style](#).

The following is the parent element of the **Map.Style** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.209.26 Map.TextAntiAliasingQuality

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.TextAntiAliasingQuality** element specifies the **anti-aliasing** quality for the text of a Map. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**High:** High anti-aliasing quality is applied to text.

**Normal:** Normal anti-aliasing quality is applied to text.

**SystemDefault:** System default anti-aliasing quality is applied to text.

If this element is not present, its value is interpreted as "High".

The following is the parent element of the **Map.TextAntiAliasingQuality** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.TextAntiAliasingQuality** element.

```
<xsd:element name="TextAntiAliasingQuality" type="xsd:string" minOccurs="0" />
```

## 2.209.27 Map.TileLanguage

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Map.TileLanguage** element specifies the primary language of **map tiles**. This element is optional.

If the **Map.TileLanguage** element is present, its value MUST be a valid language code string or an expression that evaluates to a valid language code string. The language code is a combination of the following:

- A two-letter lowercase culture code that is associated with a language, as specified in [\[ISO639-2\]](#)



- A two-letter uppercase subculture code that is associated with a country or region, as specified in [\[ISO3166-1\]](#)

If this element is not present, its value is interpreted to be the same as that of **Map.StyleLanguage**.

The following is the parent element of the **Map.TileLanguage** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.TileLanguage** element.

```
<xsd:element name="TileLanguage" type="xsd:string" minOccurs="0" />
```

## 2.209.28 Map.ToolTip

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **Map.ToolTip** element specifies the tooltip text for a map. The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **Map.ToolTip** element is optional. The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The following is the parent element of the **Map.ToolTip** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

## 2.209.29 Map.Top

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **Map.Top** element specifies the distance of a map from the top of map's container. This element is optional. If this element is present, its value MUST be an [RdlSize](#). If the **Map.Top** element is not present, its value is interpreted as 0.

The following is the parent element of the **Map.Top** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.Top** element.

```
<xsd:element name="Top" type="SizeType" minOccurs="0" />
```

## 2.209.30 Map.Visibility

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **Map.Visibility** element specifies whether a Map is hidden. This element is optional. This element is of type [Visibility](#).

The following is the parent element of the **Map.Visibility** element.

Parent elements
<a href="#">Map</a>

The following is the XML Schema definition of the **Map.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

## 2.209.31 Map.Width

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **Map.Width** element specifies the width of a [Map](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#). If this element is not present, its value is interpreted as the width of the map's container (such as a [Rectangle](#) or [Body](#)) minus the value of the peer [Map.Left](#) element, if specified.

The following is the parent element of the **Map.Width** element.

Parent elements
Map

The following is the XML Schema definition of the **Map.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="0" />
```

## 2.209.32 Map.ZIndex

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **Map.ZIndex** element specifies the drawing order of a [Map](#) within its container. This element is optional. If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17). If this element is not present, its value is interpreted as 0. The value of this element MUST be greater than or equal to 0 and less than or equal to 2147483647.

The following is the parent element of the **Map.ZIndex** element.

Parent elements
Map

The following is the XML Schema definition of the **Map.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
```

## 2.210 MapBorderSkin

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapBorderSkin** element specifies the appearance of the border around a [Map](#).

The following are the parent and child elements of the **MapBorderSkin** element.

Parent elements
<a href="#">Map</a>

Child elements
<a href="#">MapBorderSkin.MapBorderSkinType</a>
<a href="#">MapBorderSkin.Style</a>

The following is the XML Schema definition of the **MapBorderSkin** element.

```
<xsd:complexType name="MapBorderSkinType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapBorderSkinType" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.210.1 MapBorderSkin.MapBorderSkinType

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapBorderSkin.MapBorderSkinType** element specifies the **border skin** type for a Map. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**None:** No border skin is applied.

**Emboss:** An embossed border skin is applied.

**Raised:** A raised border skin is applied.

**Sunken:** A sunken border skin is applied.

**FrameThin1:** The FrameThin1 border skin is applied.

**FrameThin2:** The FrameThin2 border skin is applied.

**FrameThin3:** The FrameThin3 border skin is applied.

**FrameThin4:** The FrameThin4 border skin is applied.

**FrameThin5:** The FrameThin5 border skin is applied.

**FrameThin6:** The FrameThin6 border skin is applied.

**FrameTitle1:** The FrameTitle1 border skin is applied.

**FrameTitle2:** The FrameTitle2 border skin is applied.

**FrameTitle3:** The FrameTitle3 border skin is applied.

**FrameTitle4:** The FrameTitle4 border skin is applied.

**FrameTitle5:** The FrameTitle5 border skin is applied.

**FrameTitle6:** The FrameTitle6 border skin is applied.

**FrameTitle7:** The FrameTitle7 border skin is applied.

**FrameTitle8:** The FrameTitle8 border skin is applied.

If this element is not present, its value is interpreted as "None".

The following is the parent element of the **MapBorderSkin.MapBorderSkinType** element.

Parent elements
<a href="#">MapBorderSkin</a>

The following is the XML Schema definition of the **MapBorderSkin.MapBorderSkinType** element.

```
<xsd:element name="MapBorderSkinType" type="xsd:string" minOccurs="0" />
```

## 2.210.2 MapBorderSkin.Style

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapBorderSkin.Style** element specifies style information for a map **border skin**. This element is optional. This element is of type [Style](#).

The following is the parent element of the **MapBorderSkin.Style** element.

Parent elements
<a href="#">MapBorderSkin</a>

The following is the XML Schema definition of the **MapBorderSkin.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.211 MapColorScale

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorScale** element specifies a color scale for a [Map](#).

The following are the parent and child elements of the **MapColorScale** element.

Parent elements
Map

Child elements
<a href="#">MapColorScale.ActionInfo</a>
<a href="#">MapColorScale.BottomMargin</a>
<a href="#">MapColorScale.ColorBarBorderColor</a>
<a href="#">MapColorScale.DockOutsideViewport</a>
<a href="#">MapColorScale.Hidden</a>
<a href="#">MapColorScale.HideEndLabels</a>
<a href="#">MapColorScale.LabelBehavior</a>
<a href="#">MapColorScale.LabelFormat</a>
<a href="#">MapColorScale.LabelInterval</a>
<a href="#">MapColorScale.LabelPlacement</a>
<a href="#">MapColorScale.LeftMargin</a>
<a href="#">MapColorScale.MapColorScaleTitle</a>
<a href="#">MapColorScale.MapLocation</a>
<a href="#">MapColorScale.MapSize</a>
<a href="#">MapColorScale.NoDataText</a>
<a href="#">MapColorScale.Position</a>
<a href="#">MapColorScale.RangeGapColor</a>
<a href="#">MapColorScale.RightMargin</a>
<a href="#">MapColorScale.Style</a>
<a href="#">MapColorScale.TickMarkLength</a>
<a href="#">MapColorScale.ToolTip</a>
<a href="#">MapColorScale.TopMargin</a>
<a href="#">MapColorScale.ZIndex</a>

The following is the XML Schema definition of the **MapColorScale** element.

```

<xsd:complexType name="MapColorScaleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSubItem Start-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
    <xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
  </choice>
</complexType>

```

```

<xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
<!--MapSubItem End-->
<!--MapDockableSubItem Start-->
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
<xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<!--MapDockableSubItem End-->
<xsd:element name="MapColorScaleTitle" type="MapColorScaleTitleType"
  minOccurs="0" />
<xsd:element name="TickMarkLength" type="xsd:string" minOccurs="0" />
<xsd:element name="ColorBarBorderColor" type="xsd:string" minOccurs="0" />
<xsd:element name="LabelInterval" type="xsd:string" minOccurs="0" />
<xsd:element name="LabelFormat" type="xsd:string" minOccurs="0" />
<xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
<xsd:element name="LabelBehavior" type="xsd:string" minOccurs="0" />
<xsd:element name="HideEndLabels" type="xsd:string" minOccurs="0" />
<xsd:element name="RangeGapColor" type="xsd:string" minOccurs="0" />
<xsd:element name="NoDataText" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.211.1 MapColorScale.ActionInfo

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorScale.ActionInfo** element specifies a collection of actions for a **map color scale**. This element is optional. This element is of type [ActionInfo](#).

The following is the parent element of the **MapColorScale.ActionInfo** element.

Parent elements
<a href="#">MapColorScale</a>

The following is the XML Schema definition of the **MapColorScale.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.211.2 MapColorScale.BottomMargin

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorScale.BottomMargin** element specifies the bottom margin for a [MapColorScale](#) and its parent [Map](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **MapColorScale.BottomMargin** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.BottomMargin** element.

```
<xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
```

### 2.211.3 MapColorScale.ColorBarBorderColor

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorScale.ColorBarBorderColor** element specifies the border color of the color bar for a [MapColorScale](#). The **MapColorScale.ColorBarBorderColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. If this element is not present, its value is interpreted as Black.

The following is the parent element of the **MapColorScale.ColorBarBorderColor** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.ColorBarBorderColor** element.

```
<xsd:element name="ColorBarBorderColor" type="xsd:string" minOccurs="0" />
```

### 2.211.4 MapColorScale.DockOutsideViewport

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorScale.DockOutsideViewport** element specifies whether a [MapColorScale](#) will be docked outside a [MapViewport](#). This element is optional.

If the **MapColorScale.DockOutsideViewport** element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false. This element is ignored if the value of the [MapColorScale.MapLocation](#) element is defined.

The following is the parent element of the **MapColorScale.DockOutsideViewport** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.DockOutsideViewport** element.

```
<xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
```

### 2.211.5 MapColorScale.Hidden

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorScale.Hidden** element specifies whether a [MapColorScale](#) will be hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **MapColorScale.Hidden** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

### 2.211.6 MapColorScale.HideEndLabels

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorScale.HideEndLabels** element specifies whether the end labels for a [MapColorScale](#) will be hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

The following is the parent element of the **MapColorScale.HideEndLabels** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.HideEndLabels** element.

```
<xsd:element name="HideEndLabels" type="xsd:string" minOccurs="0" />
```

### 2.211.7 MapColorScale.LabelBehavior

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorScale.LabelBehavior** element specifies how the labels display the values in a [MapColorScale](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Auto:** The value is automatically determined.

**ShowMiddleValue:** The middle value is shown.

**ShowBorderValue:** The border value is shown.

If this element is not present, its value is interpreted as "Auto".

The following is the parent element of the **MapColorScale.LabelBehavior** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.LabelBehavior** element.



```
<xsd:element name="LabelBehavior" type="xsd:string" minOccurs="0" />
```

### 2.211.8 MapColorScale.LabelFormat

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorScale.LabelFormat** element specifies the label format for a [MapColorScale](#). This element is optional. If the **MapColorScale.LabelFormat** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as `#,##0.##`.

The following is the parent element of the **MapColorScale.LabelFormat** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.LabelFormat** element.

```
<xsd:element name="LabelFormat" type="xsd:string" minOccurs="0" />
```

### 2.211.9 MapColorScale.LabelInterval

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorScale.LabelInterval** element specifies the interval between labels in a [MapColorScale](#). This element is optional. If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 1.

The following is the parent element of the **MapColorScale.LabelInterval** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.LabelInterval** element.

```
<xsd:element name="LabelInterval" type="xsd:string" minOccurs="0" />
```

### 2.211.10 MapColorScale.LabelPlacement

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorScale.LabelPlacement** element specifies the placement of labels in a [MapColorScale](#). This element is optional. If the **MapColorScale.LabelPlacement** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Alternate:** Odd labels are positioned below the color bar, and even labels are positioned above the color bar.

**Top:** All labels are positioned above the color bar.

**Bottom:** All labels are positioned below the color bar.

If this element is not present, its value is interpreted as "Alternate".

The following is the parent element of the **MapColorScale.LabelPlacement** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.LabelPlacement** element.

```
<xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
```

### 2.211.11 MapColorScale.LeftMargin

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorScale.LeftMargin** element specifies the left margin for a [MapColorScale](#) and its parent [Map](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **MapColorScale.LeftMargin** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.LeftMargin** element.

```
<xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
```

### 2.211.12 MapColorScale.MapColorScaleTitle

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorScale.MapColorScaleTitle** element specifies a title for a [MapColorScale](#). This element is optional. This element is of type [MapColorScaleTitle](#).

The following is the parent element of the **MapColorScale.MapColorScaleTitle** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.MapColorScaleTitle** element.

```
<xsd:element name="MapColorScaleTitle" type="MapColorScaleTitleType"
  minOccurs="0" />
```

### 2.211.13 MapColorScale.MapLocation

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorScale.MapLocation** element specifies the location of a [MapColorScale](#). This element is optional. If this element is present, the [MapColorScale.Position](#) element is ignored. This element is of type [MapLocation](#).

The following is the parent element of the **MapColorScale.MapLocation** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.MapLocation** element.

```
<xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
```

### 2.211.14 MapColorScale.MapSize

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorScale.MapSize** element specifies the size of a [MapColorScale](#). This element is optional. If this element is not present, the color scale size MUST be automatically calculated. This element is of type [MapSize](#).

The following is the parent element of the **MapColorScale.MapSize** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.MapSize** element.

```
<xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
```

### 2.211.15 MapColorScale.NoDataText

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorScale.NoDataText** element specifies the label for the [MapColorScale](#) that is to be used with colors that have no data associated with them. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

The following is the parent element of the **MapColorScale.NoDataText** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.NoDataText** element.

```
<xsd:element name="NoDataText" type="xsd:string" minOccurs="0" />
```

## 2.211.16 MapColorScale.Position

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorScale.Position** element specifies the position of a [MapColorScale](#) in a [Map](#). This element is optional.

The **MapColorScale.Position** element is ignored if a value for the [MapColorScale.MapLocation](#) element is specified. If the **MapColorScale.Position** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**TopCenter:** The color scale is positioned at the top center.

**TopLeft:** The color scale is positioned at the top left.

**TopRight:** The color scale is positioned at the top right.

**LeftTop:** The color scale is positioned at the left top.

**LeftCenter:** The color scale is positioned at the left center.

**LeftBottom:** The color scale is positioned at the left bottom.

**RightTop:** The color scale is positioned at the right top.

**RightCenter:** The color scale is positioned at the right center.

**RightBottom:** The color scale is positioned at the right bottom.

**BottomRight:** The color scale is positioned at the bottom right.

**BottomCenter:** The color scale is positioned at the bottom center.

**BottomLeft:** The color scale is positioned at the bottom left.

If this element is not present, its value is interpreted as "TopCenter".

The following is the parent element of the **MapColorScale.Position** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.Position** element.

```
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
```

## 2.211.17 MapColorScale.RangeGapColor

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorScale.RangeGapColor** element specifies the color to be used in a [MapColorScale](#) to fill the undefined color divisions. This element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. If this element is not present, its value is interpreted as "White".

The following is the parent element of the **MapColorScale.RangeGapColor** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.RangeGapColor** element.

```
<xsd:element name="RangeGapColor" type="xsd:string" minOccurs="0" />
```

### 2.211.18 MapColorScale.RightMargin

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorScale.RightMargin** element specifies the right margin for a [MapColorScale](#) and its parent [Map](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **MapColorScale.RightMargin** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.RightMargin** element.

```
<xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
```

### 2.211.19 MapColorScale.Style

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorScale.Style** element specifies style information for a [MapColorScale](#). The **MapColorScale.Style** element is optional. This element is of type [Style](#).

The following is the parent element of the **MapColorScale.Style** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.211.20 MapColorScale.TickMarkLength

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorScale.TickMarkLength** element specifies the length of the tick mark in a [MapColorScale](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 2.25pt.

The following is the parent element of the **MapColorScale.TickMarkLength** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.TickMarkLength** element.

```
<xsd:element name="TickMarkLength" type="xsd:string" minOccurs="0" />
```

### 2.211.21 MapColorScale.ToolTip

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorScale.ToolTip** element specifies the tooltip text for a [MapColorScale](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **MapColorScale.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

The following is the parent element of the **MapColorScale.ToolTip** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

### 2.211.22 MapColorScale.TopMargin

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorScale.TopMargin** element specifies the top margin for a [MapColorScale](#) and its parent [Map](#). This element is optional.

If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **MapColorScale.TopMargin** element.

Parent elements
MapColorScale

The following is the XML Schema definition of the **MapColorScale.TopMargin** element.

```
<xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
```

### 2.211.23 MapColorScale.ZIndex

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorScale.ZIndex** element specifies the drawing order of a [MapColorScale](#) within its container. This element is optional.

If the **MapColorScale.ZIndex** element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **MapColorScale.ZIndex** element.

Parent elements
<a href="#">MapColorScale</a>

The following is the XML Schema definition of the **MapColorScale.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
```

## 2.212 MapColorScaleTitle

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorScaleTitle** element specifies the title of a [MapColorScale](#).

The following are the parent and child elements of the **MapColorScaleTitle** element.

Parent elements
<a href="#">MapColorScale</a>

Child elements
<a href="#">MapColorScaleTitle.Caption</a>
<a href="#">MapColorScaleTitle.Style</a>

The following is the XML Schema definition of the **MapColorScaleTitle** element.

```
<xsd:complexType name="MapColorScaleTitleType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Caption" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.212.1 MapColorScaleTitle.Caption

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorScaleTitle.Caption** element specifies the caption for the title of a [MapColorScale](#). This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

The following is the parent element of the **MapColorScaleTitle.Caption** element.

Parent elements
<a href="#">MapColorScaleTitle</a>

The following is the XML Schema definition of the **MapColorScaleTitle.Caption** element.

```
<xsd:element name="Caption" type="xsd:string" minOccurs="0" />
```

### 2.212.2 MapColorScaleTitle.Style

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapColorScaleTitle.Style** element specifies style information for the [MapColorScaleTitle](#) element. The **MapColorScaleTitle.Style** element is optional. This element is of type [Style](#).

The following is the parent element of the **MapColorScaleTitle.Style** element.

Parent elements
MapColorScaleTitle

The following is the XML Schema definition of the **MapColorScaleTitle.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.213 MapLocation

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapLocation** element specifies the location of a [MapViewport](#), a [MapTitle](#), a [MapLegend](#), a [MapDistanceScale](#), or a [MapColorScale](#).

The following are the parent and child elements of the **MapLocation** element.

Parent elements
MapColorScale
MapDistanceScale
MapLegend
MapTitle
MapViewport

Child elements
<a href="#">MapLocation.Left</a>
<a href="#">MapLocation.Top</a>



<b>Child elements</b>
-----------------------

<a href="#">MapLocation.Unit</a>
----------------------------------

The following is the XML Schema definition of the **MapLocation** element.

```
<xsd:complexType name="MapLocationType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Unit" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.213.1 MapLocation.Left

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLocation.Left** element specifies the distance from the left of a [MapViewport](#), a [MapTitle](#), a [MapLegend](#), a [MapDistanceScale](#), and a [MapColorScale](#) relative to the containing [Map](#) or **MapViewport**. The unit of measure is specified by the [MapLocation.Unit](#) element. The **MapLocation.Left** element is optional.

If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **MapLocation.Left** element.

<b>Parent elements</b>
------------------------

<a href="#">MapLocation</a>
-----------------------------

The following is the XML Schema definition of the **MapLocation.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

### 2.213.2 MapLocation.Top

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLocation.Top** element specifies the distance from the top of a [MapViewport](#), a [MapTitle](#), a [MapLegend](#), a [MapDistanceScale](#), and a [MapColorScale](#) relative to the containing [Map](#) or **MapViewport**. The unit of measure is specified by the [MapLocation.Unit](#) element. The **MapLocation.Top** element is optional.

If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

The following is the parent element of the **MapLocation.Top** element.

<b>Parent elements</b>
------------------------

<a href="#">MapLocation</a>
-----------------------------

The following is the XML Schema definition of the **MapLocation.Top** element.

```
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
```

### 2.213.3 MapLocation.Unit

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLocation.Unit** element specifies the unit of measure to be used for both [MapLocation.Left](#) and [MapLocation.Top](#). The **MapLocation.Unit** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Percentage:** Specifies that the unit is a percentage of the containing object.

**Inch:** Specifies that the unit is inches.

**Point:** Specifies that the unit is points.

**Centimeter:** Specifies that the unit is centimeters.

**Millimeter:** Specifies that the unit is millimeters.

**Pica:** Specifies that the unit is picas.

If this element is not present, its value is interpreted as "Percentage".

The following is the parent element of the **MapLocation.Unit** element.

Parent elements
<a href="#">MapLocation</a>

The following is the XML Schema definition of the **MapLocation.Unit** element.

```
<xsd:element name="Unit" type="xsd:string" minOccurs="0" />
```

### 2.214 MapSize

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapSize** element specifies the size of a [MapViewport](#), [MapTitle](#), [MapLegend](#), [MapDistanceScale](#), or [MapColorScale](#).

The following are the parent and child elements of the **MapSize** element.

Parent elements
MapColorScale
MapDistanceScale
MapLegend
MapTitle
MapViewport

<b>Child elements</b>
<a href="#">MapSize.Height</a>
<a href="#">MapSize.Unit</a>
<a href="#">MapSize.Width</a>

The following is the XML Schema definition of the **MapSize** element.

```
<xsd:complexType name="MapSizeType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Width" type="xsd:string" minOccurs="1" />
    <xsd:element name="Height" type="xsd:string" minOccurs="1" />
    <xsd:element name="Unit" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.214.1 MapSize.Height

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapSize.Height** element specifies the height of a [MapViewport](#), [MapTitle](#), [MapLegend](#), [MapDistanceScale](#), or [MapColorScale](#) relative to the containing [Map](#) or **MapViewport**. The **MapSize.Height** element MUST be specified.

The value of this element MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. The unit of height is specified by the [MapSize.Unit](#) element.

Following is the parent element of the **MapSize.Height** element.

<b>Parent elements</b>
<a href="#">MapSize</a>

The following is the XML Schema definition of the **MapSize.Height** element.

```
<xsd:element name="Height" type="xsd:string" minOccurs="1" />
```

### 2.214.2 MapSize.Unit

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapSize.Unit** element specifies the unit of measure to be use for both the [MapSize.Height](#) element and the [MapSize.Width](#) element. The **MapSize.Unit** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Percentage:** Specifies that the unit is a percentage of the containing object.

**Inch:** Specifies that the unit is inches.

**Point:** Specifies that the unit is points.

**Centimeter:** Specifies that the unit is centimeters.

**Millimeter:** Specifies that the unit is millimeters.

**Pica:** Specifies that the unit is picas.

If this element is not present, its value is interpreted as "Percentage".

Following is the parent element of the **MapSize.Unit** element.

Parent elements
<a href="#">MapSize</a>

The following is the XML Schema definition of the **MapSize.Unit** element.

```
<xsd:element name="Unit" type="xsd:string" minOccurs="0" />
```

### 2.214.3 MapSize.Width

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSize.Width** element specifies the width of a [MapViewport](#), [MapTitle](#), [MapLegend](#), [MapDistanceScale](#), or [MapColorScale](#) relative to the containing [Map](#) or **MapViewport**. The **MapSize.Width** element MUST be specified.

The value of this element MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. The unit of width is specified by the [MapSize.Unit](#) element.

Following is the parent element of the **MapSize.Width** element.

Parent elements
<a href="#">MapSize</a>

The following is the XML Schema definition of the **MapSize.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="1" />
```

### 2.215 MapDataRegions

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDataRegions** element specifies a collection of [MapDataRegion](#) elements. The **MapDataRegions** element MUST contain at least one **MapDataRegion** element.

The following are the parent and child elements of the **MapDataRegions** element.

Parent elements
<a href="#">Map</a>

<b>Child elements</b>
-----------------------

<a href="#">MapDataRegions.MapDataRegion</a>
--

The following is the XML Schema definition of the **MapDataRegions** element.

```
<xsd:complexType name="MapDataRegionsType">
  <xsd:sequence>
    <xsd:element name="MapDataRegion" type="MapDataRegionType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.215.1 MapDataRegions.MapDataRegion

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDataRegions.MapDataRegion** element specifies a **data region** for a map. This element **MUST** be specified. This element is of type [MapDataRegion](#).

Following is the parent element of the **MapDataRegions.MapDataRegion** element.

<b>Parent elements</b>
------------------------

<a href="#">MapDataRegions</a>
--------------------------------

The following is the XML Schema definition of the **MapDataRegions.MapDataRegion** element.

```
<xsd:element name="MapDataRegion" type="MapDataRegionType" minOccurs="1"
  maxOccurs="unbounded" />
```

### 2.216 MapDataRegion

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDataRegion** element provides a **map polygon layer**, a map **point layer**, or a map **line layer** with analytical data after it applies filtering and/or grouping to the layer.

The following are the parent elements, attributes, and child elements of the **MapDataRegion** element.

<b>Parent elements</b>
------------------------

<a href="#">MapDataRegions</a>
--------------------------------

<b>Attributes</b>
-------------------

<a href="#">MapDataRegion.Name</a>
------------------------------------

Child elements
<a href="#">MapDataRegion.DataSetName</a>
<a href="#">MapDataRegion.Filters</a>
<a href="#">MapDataRegion.MapMember</a>

**Applies to** [RDL 2011/01](#)

Child elements
<a href="#">MapDataRegion.Relationship</a>

The following is the XML Schema definition of the **MapDataRegion** element.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:complexType name="MapDataRegionType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="MapMember" type="MapMemberType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.216.1 MapDataRegion.Name

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapDataRegion.Name** attribute specifies the name of a [MapDataRegion](#). This attribute MUST be specified. The value of this attribute MUST be a case-sensitive [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#).

Following is the parent element of the **MapDataRegion.Name** attribute.

Parent elements
MapDataRegion

The following is the XML Schema definition of the **MapDataRegion.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.216.2 MapDataRegion.DataSetName

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapDataRegion.DataSetName** element specifies the name of the dataset to use for a [MapDataRegion](#). The **MapDataRegion.DataSetName** element is optional.

This element MUST be specified if the following conditions are true:

- The map is not contained in another data region.
- More than one dataset is specified in the report.

If the **MapDataRegion** has an ancestor, the value of the **MapDataRegion.DataSetName** element is interpreted as the **DataSet.Name** for the containing scope (**DataRegion**, [Group](#), or **Cell**).<56>

The **MapDataRegion.DataSetName** element is ignored if the map is contained in another data region. If this element is not present and the map is not contained in another data region, the value of the **MapDataRegion.DataSetName** element is interpreted as the name of the single dataset name defined in the report. If the **MapDataRegion.DataSetName** element is contained in another data region, its value is interpreted as the name of the dataset of the containing data region.

Following is the parent element of the **MapDataRegion.DataSetName** element.

Parent elements
<a href="#">MapDataRegion</a>

The following is the XML Schema definition of the **MapDataRegion.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
```

### 2.216.3 MapDataRegion.Filters

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDataRegion.Filters** element specifies a collection of filters for a Map. This element is optional. This element is of type [Filters](#).

Following is the parent element of the **MapDataRegion.Filters** element.

Parent elements
<a href="#">MapDataRegion</a>

The following is the XML Schema definition of the **MapDataRegion.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

### 2.216.4 MapDataRegion.MapMember

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDataRegion.MapMember** element specifies the grouping and/or the filter to apply to the containing [MapDataRegion](#). This element is optional. This element is of type [MapMember](#).

Following is the parent element of the **MapDataRegion.MapMember** element.

Parent elements
<a href="#">MapDataRegion</a>

The following is the XML Schema definition of the **MapDataRegion.MapMember** element.

```
<xsd:element name="MapMember" type="MapMemberType" minOccurs="0" />
```

## 2.216.5 MapDataRegion.Relationship

**Applies to** [RDL 2011/01](#)

The **MapDataRegion.Relationship** element specifies a **relationship** to use for correlating data in a [MapDataRegion](#) with the data in the containing scope. The **MapDataRegion.Relationship** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [Relationship](#). The **MapDataRegion.Relationship** element is ignored if the dataset for this **MapDataRegion** is the same as the dataset for each containing scope. The **MapDataRegion.Relationship** element MUST NOT be specified if there is no containing scope.

Following is the parent element of the **MapDataRegion.Relationship** element.

Parent elements
<a href="#">MapDataRegion</a>

The following is the XML Schema definition of the **MapDataRegion.Relationship** element.

```
<xsd:element name="Relationship" type="RelationshipType" minOccurs="0" />
```

## 2.217 MapMember

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapMember** element specifies group and filter behavior for the data in a [MapDataRegion](#) element or in the parent **MapMember** element.

The following are the parent and child elements of the **MapMember** element.

Parent elements
<a href="#">MapDataRegion</a>
<a href="#">MapMember.MapMember</a>

Child elements
<a href="#">MapMember.Group</a>
<a href="#">MapMember.MapMember</a>

The following is the XML Schema definition of the **MapMember** element.

```
<xsd:complexType name="MapMemberType">  
  <xsd:choice maxOccurs="unbounded">  
    <xsd:element name="Group" type="GroupType" minOccurs="1" />  
    <xsd:element name="MapMember" type="MapMemberType" minOccurs="0" />  
  </xsd:choice>  
</xsd:complexType>
```



```

    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.217.1 MapMember.Group

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMember.Group** element specifies grouping and filtering to apply to the data in a [MapMember](#). This element MUST be specified. The **MapMember.Group** element is of type [Group](#).

Following is the parent element of the **MapMember.Group** element.

Parent elements
<a href="#">MapMember</a>

The following is the XML Schema definition of the **MapMember.Group** element.

```
<xsd:element name="Group" type="GroupType" minOccurs="1" />
```

### 2.217.2 MapMember.MapMember

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMember.MapMember** element specifies nested grouping or filtering behavior for data in a map member. This element is optional. The **MapMember.MapMember** element is of type [MapMember](#).

Following is the parent element of the **MapMember.MapMember** element.

Parent elements
<a href="#">MapMember</a>

The following is the XML Schema definition of the **MapMember.MapMember** element.

```
<xsd:element name="MapMember" type="MapMemberType" minOccurs="0" />
```

### 2.218 MapDistanceScale

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDistanceScale** element specifies the **map distance scale**.

The following are the parent and child elements of the **MapDistanceScale** element.

Parent elements
<a href="#">Map</a>

Child elements
<a href="#">MapDistanceScale.ActionInfo</a>
<a href="#">MapDistanceScale.BottomMargin</a>
<a href="#">MapDistanceScale.DockOutsideViewport</a>
<a href="#">MapDistanceScale.Hidden</a>
<a href="#">MapDistanceScale.LeftMargin</a>
<a href="#">MapDistanceScale.MapLocation</a>
<a href="#">MapDistanceScale.MapSize</a>
<a href="#">MapDistanceScale.Position</a>
<a href="#">MapDistanceScale.RightMargin</a>
<a href="#">MapDistanceScale.ScaleBorderColor</a>
<a href="#">MapDistanceScale.ScaleColor</a>
<a href="#">MapDistanceScale.Style</a>
<a href="#">MapDistanceScale.ToolTip</a>
<a href="#">MapDistanceScale.TopMargin</a>
<a href="#">MapDistanceScale.ZIndex</a>

The following is the XML Schema definition of the **MapDistanceScale** element.

```

<xsd:complexType name="MapDistanceScaleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSubItem Start-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
    <xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <!--MapSubItem End-->
    <!--MapDockableSubItem Start-->
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <!--MapDockableSubItem End-->
    <xsd:element name="ScaleColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ScaleBorderColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

## 2.218.1 MapDistanceScale.ActionInfo

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapDistanceScale.ActionInfo** element specifies a set of actions for a [MapDistanceScale](#). The **MapDistanceScale.ActionInfo** element is optional. This element is of type [ActionInfo](#).

Following is the parent element of the **MapDistanceScale.ActionInfo** element.

Parent elements
MapDistanceScale

The following is the XML Schema definition of the **MapDistanceScale.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

## 2.218.2 MapDistanceScale.BottomMargin

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDistanceScale.BottomMargin** element specifies the bottom margin for a [MapDistanceScale](#) and its parent [Map](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapDistanceScale.BottomMargin** element.

Parent elements
MapDistanceScale

The following is the XML Schema definition of the **MapDistanceScale.BottomMargin** element.

```
<xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
```

## 2.218.3 MapDistanceScale.DockOutsideViewport

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDistanceScale.DockOutsideViewport** element specifies whether a [MapDistanceScale](#) will be docked outside a [MapViewport](#). This element is optional.

If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/21](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false. This element is ignored if the value of the [MapColorScale.MapLocation](#) element is defined.

Following is the parent element of the **MapDistanceScale.DockOutsideViewport** element.

Parent elements
MapDistanceScale

The following is the XML Schema definition of the **MapDistanceScale.DockOutsideViewport** element.

```
<xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
```

## 2.218.4 MapDistanceScale.Hidden

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapDistanceScale.Hidden** element specifies whether a [MapDistanceScale](#) will be hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapDistanceScale.Hidden** element.

Parent elements
MapDistanceScale

The following is the XML Schema definition of the **MapDistanceScale.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

## 2.218.5 MapDistanceScale.LeftMargin

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapDistanceScale.LeftMargin** element specifies the left margin for a [MapDistanceScale](#) and its parent [Map](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapDistanceScale.LeftMargin** element.

Parent elements
MapDistanceScale

The following is the XML Schema definition of the **MapDistanceScale.LeftMargin** element.

```
<xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
```

## 2.218.6 MapDistanceScale.MapLocation

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapDistanceScale.MapLocation** element specifies the location of a [MapDistanceScale](#). This element is optional. If this element is present, the value of the [MapDistanceScale.Position](#) element is ignored. The **MapDistanceScale.MapLocation** element is of type [MapLocation](#).

Following is the parent element of the **MapDistanceScale.MapLocation** element.

Parent elements
MapDistanceScale

The following is the XML Schema definition of the **MapDistanceScale.MapLocation** element.

```
<xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
```

## 2.218.7 MapDistanceScale.MapSize

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDistanceScale.MapSize** element specifies the size of a [MapDistanceScale](#). This element is optional. If this element is not present, the size of the distance scale **MUST** be automatically calculated. This element is of type [MapSize](#).

Following is the parent element of the **MapDistanceScale.MapSize** element.

Parent elements
MapDistanceScale

The following is the XML Schema definition of the **MapDistanceScale.MapSize** element.

```
<xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
```

## 2.218.8 MapDistanceScale.Position

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDistanceScale.Position** element specifies the position of a [MapDistanceScale](#) in a Map. This element is optional.

This element is ignored if the value of the **MapDistanceScale.Position** element is specified. If the **MapDistanceScale.Position** element is present, its value **MUST** be a **String** or an expression that evaluates to a **String**.

The value of this element **MUST** be one of the following or an expression that evaluates to one of the following:

**TopCenter:** The distance scale is positioned at the top center.

**TopLeft:** The distance scale is positioned at the top left.

**TopRight:** The distance scale is positioned at the top right.

**LeftTop:** The distance scale is positioned at the left top.

**LeftCenter:** The distance scale is positioned at the left center.

**LeftBottom:** The distance scale is positioned at the left bottom.

**RightTop:** The distance scale is positioned at the right top.

**RightCenter:** The distance scale is positioned at the right center.

**RightBottom:** The distance scale is positioned at the right bottom.

**BottomRight:** The distance scale is positioned at the bottom right.

**BottomCenter:** The distance scale is positioned at the bottom center.

**BottomLeft:** The distance scale is positioned at the bottom left.

If this element is not present, its value is interpreted as "TopCenter".

Following is the parent element of the **MapDistanceScale.Position** element.

Parent elements
MapDistanceScale

The following is the XML Schema definition of the **MapDistanceScale.Position** element.

```
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
```

### 2.218.9 MapDistanceScale.RightMargin

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDistanceScale.RightMargin** element specifies the right margin of a [MapDistanceScale](#) and its parent [Map](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapDistanceScale.RightMargin** element.

Parent elements
MapDistanceScale

The following is the XML Schema definition of the **MapDistanceScale.RightMargin** element.

```
<xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
```

### 2.218.10 MapDistanceScale.ScaleBorderColor

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDistanceScale.ScaleBorderColor** element specifies the scale border color in a [MapDistanceScale](#). This element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. If this element is not present, its value is interpreted as DarkGray.

Following is the parent element of the **MapDistanceScale.ScaleBorderColor** element.

Parent elements
MapDistanceScale

The following is the XML Schema definition of the **MapDistanceScale.ScaleBorderColor** element.

```
<xsd:element name="ScaleBorderColor" type="xsd:string" minOccurs="0" />
```

### 2.218.11 MapDistanceScale.ScaleColor

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDistanceScale.ScaleColor** element specifies the scale color in a [MapDistanceScale](#). This element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. If this element is not present, its value is interpreted as White.

Following is the parent element of the **MapDistanceScale.ScaleColor** element.

Parent elements
MapDistanceScale

The following is the XML Schema definition of the **MapDistanceScale.ScaleColor** element.

```
<xsd:element name="ScaleColor" type="xsd:string" minOccurs="0" />
```

### 2.218.12 MapDistanceScale.Style

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDistanceScale.Style** specifies style information for a [MapDistanceScale](#). This element is optional. This element is of type [Style](#).

Following is the parent element of the **MapDistanceScale.Style** element.

Parent elements
MapDistanceScale

The following is the XML Schema definition of the **MapDistanceScale.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.218.13 MapDistanceScale.ToolTip

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDistanceScale.ToolTip** element specifies the tooltip text for a [MapDistanceScale](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **MapDistanceScale.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **MapDistanceScale.ToolTip** element.

Parent elements
MapDistanceScale

The following is the XML Schema definition of the **MapDistanceScale.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

## 2.218.14 MapDistanceScale.TopMargin

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapDistanceScale.TopMargin** element specifies the top margin for a [MapDistanceScale](#) and its parent [Map](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapDistanceScale.TopMargin** element.

Parent elements
<a href="#">MapDistanceScale</a>

The following is the XML Schema definition of the **MapDistanceScale.TopMargin** element.

```
<xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
```

## 2.218.15 MapDistanceScale.ZIndex

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapDistanceScale.ZIndex** element specifies the drawing order of a [MapDistanceScale](#) within its container. This element is optional. If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapDistanceScale.ZIndex** element.

Parent elements
<a href="#">MapDistanceScale</a>

The following is the XML Schema definition of the **MapDistanceScale.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
```

## 2.219 MapLayers

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLayers** element specifies a collection of [MapTileLayer](#), [MapLineLayer](#), [MapPointLayer](#), and [MapPolygonLayer](#) elements.

The following are the parent and child elements of the **MapLayers** element.

Parent elements
<a href="#">Map</a>



Child elements
MapLineLayer
MapPointLayer
MapPolygonLayer
MapTileLayer

The following is the XML Schema definition of the **MapLayers** element.

```
<xsd:complexType name="MapLayersType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapTileLayer" type="MapTileLayerType" />
    <xsd:element name="MapPolygonLayer" type="MapPolygonLayerType" />
    <xsd:element name="MapPointLayer" type="MapPointLayerType" />
    <xsd:element name="MapLineLayer" type="MapLineLayerType" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.219.1 MapLayers.MapLineLayer

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLayers.MapLineLayer** element specifies a **line layer** for a map. This element is of type [MapLineLayer](#).

Following is the parent element of the **MapLayers.MapLineLayer** element.

Parent elements
<a href="#">MapLayers</a>

The following is the XML Schema definition of the **MapLayers.MapLineLayer** element.

```
<xsd:element name="MapLineLayer" type="MapLineLayerType" />
```

### 2.219.2 MapLayers.MapPointLayer

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLayers.MapPointLayer** element specifies a **point layer** for a [Map](#). This element is of type [MapPointLayer](#).

Following is the parent element of the **MapLayers.MapPointLayer** element.

Parent elements
<a href="#">MapLayers</a>

The following is the XML Schema definition of the **MapLayers.MapPointLayer** element.

```
<xsd:element name="MapPointLayer" type="MapPointLayerType" />
```

### 2.219.3 MapLayers.MapPolygonLayer

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLayers.MapPolygonLayer** element specifies a **polygon layer** for a [Map](#). This element is of type [MapPolygonLayer](#).

Following is the parent element of the **MapLayers.MapPolygonLayer** element.

Parent elements
<a href="#">MapLayers</a>

The following is the XML Schema definition of the **MapLayers.MapPolygonLayer** element.

```
<xsd:element name="MapPolygonLayer" type="MapPolygonLayerType" />
```

### 2.219.4 MapLayers.MapTileLayer

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLayers.MapTileLayer** element specifies a **tile layer** for a [Map](#). This element is of type [MapTileLayer](#).

Following is the parent element of the **MapLayers.MapTileLayer** element.

Parent elements
<a href="#">MapLayers</a>

The following is the XML Schema definition of the **MapLayers.MapTileLayer** element.

```
<xsd:element name="MapTileLayer" type="MapTileLayerType" />
```

### 2.220 MapLineLayer

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLineLayer** element specifies a **line-based map layer** (for example, a street, a river, or a path) to be drawn in a **map**.

The following are the parent elements, attributes, and the child elements of the **MapLineLayer** element.

Parent elements
<a href="#">MapLayers</a>

Attributes
<a href="#">MapLineLayer.Name</a>

Child elements
<a href="#">MapLineLayer.DataElementName</a>
<a href="#">MapLineLayer.DataElementOutput</a>
<a href="#">MapLineLayer.MapBindingFieldPairs</a>
<a href="#">MapLineLayer.MapDataRegionName</a>
<a href="#">MapLineLayer.MapFieldDefinitions</a>
<a href="#">MapLineLayer.MapLineRules</a>
<a href="#">MapLineLayer.MapLines</a>
<a href="#">MapLineLayer.MapLineTemplate</a>
<a href="#">MapLineLayer.MapShapefile</a>
<a href="#">MapLineLayer.MapSpatialDataSet</a>
<a href="#">MapLineLayer.MapSpatialDataRegion</a>
<a href="#">MapLineLayer.MaximumZoom</a>
<a href="#">MapLineLayer.MinimumZoom</a>
<a href="#">MapLineLayer.Transparency</a>
<a href="#">MapLineLayer.VisibilityMode</a>

The following is the XML Schema definition of the **MapLineLayer** element.

```

<xsd:complexType name="MapLineLayerType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapLayerStart-->
    <xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <!--MapLayerEnd-->
    <!--MapVectorLayerStart-->
    <xsd:element name="MapDataRegionName" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType"
      minOccurs="0" />
    <xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType"
      minOccurs="0" />
    <xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
    <xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType"
      minOccurs="0" />
    <xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType"
      minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>

```

```

        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapVectorLayerEnd-->
    <xsd:element name="MapLineTemplate" type="MapLineTemplateType" minOccurs="0" />
    <xsd:element name="MapLineRules" type="MapLineRulesType" minOccurs="0" />
    <xsd:element name="MapLines" type="MapLinesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.220.1 MapLineLayer.Name

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineLayer.Name** attribute specifies a name for the [MapLineLayer](#) element. The **MapLineLayer.Name** attribute MUST be specified. The value of this attribute MUST be a case-sensitive [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#).

Following is the parent element of the **MapLineLayer.Name** attribute.

Parent elements
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.220.2 MapLineLayer.DataElementName

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineLayer.DataElementName** element specifies the name for the [MapLineLayer](#) to use for the data element or attribute. This element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as the name attribute of the **line layer**.

Following is the parent element of the **MapLineLayer.DataElementName** element.

Parent elements
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.220.3 MapLineLayer.DataElementOutput

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineLayer.DataElementOutput** element specifies whether a [MapLineLayer](#) will appear in a **data rendering**. This element is optional.

If this element is present, its value MUST be one of the following:

**Output:** Specifies that the item appears in the data rendering output.

**NoOutput:** Specifies that the item does not appear in the data rendering output.

If this element is not present, its value is interpreted as "Output".

Following is the parent element of the **MapLineLayer.DataElementOutput** element.

Parent elements
<a href="#">MapLineLayer</a>

The following is the XML Schema definition of the **MapLineLayer.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.220.4 MapLineLayer.MapBindingFieldPairs

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineLayer.MapBindingFieldPairs** element specifies a collection of [MapBindingFieldPair](#) elements to be used to bind the **map items** in the parent map **line layer** to the data region that is associated with that layer. The **MapLineLayer.MapBindingFieldPairs** element is optional.

This element MUST be specified if the value of the [MapLineLayer.MapDataRegionName](#) element is specified and the value of the [MapLineLayer.MapSpatialDataRegion](#) element is not specified. Otherwise, the value of the **MapLineLayer.MapBindingFieldPairs** element is ignored. This element is of type [MapBindingFieldPairs](#).

Following is the parent element of the **MapLineLayer.MapBindingFieldPairs** element.

Parent elements
<a href="#">MapLineLayer</a>

The following is the XML Schema definition of the **MapLineLayer.MapBindingFieldPairs** element.

```
<xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType" minOccurs="0" />
```

## 2.220.5 MapLineLayer.MapDataRegionName

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineLayer.MapDataRegionName** element specifies the data region from which to consume data. This element is optional. This element MUST be specified if the value of the [MapLineLayer.MapSpatialDataRegion](#) element is specified.

Following is the parent element of the **MapLineLayer.MapDataRegionName** element.

Parent elements
<a href="#">MapLineLayer</a>

The following is the XML Schema definition of the **MapLineLayer.MapDataRegionName** element.

```
<xsd:element name="MapDataRegionName" type="xsd:string" minOccurs="0" />
```

## 2.220.6 MapLineLayer.MapFieldDefinitions

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLineLayer.MapFieldDefinitions** element specifies a collection of [MapFieldDefinition](#) elements in a parent **line layer**. This collection describes the metadata for the fields that are associated with the **map lines**. The **MapLineLayer.MapFieldDefinitions** element is optional.

This element is ignored if the value of the [MapLineLayer.MapLines](#) element is not specified. The **MapLineLayer.MapFieldDefinitions** element is of type [MapFieldDefinitions](#).

Following is the parent element of the **MapLineLayer.MapFieldDefinitions** element.

Parent elements
<a href="#">MapLineLayer</a>

The following is the XML Schema definition of the **MapLineLayer.MapFieldDefinitions** element.

```
<xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType" minOccurs="0" />
```

## 2.220.7 MapLineLayer.MapLineRules

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLineLayer.MapLineRules** element specifies a set of rules to be applied to **map lines** in the parent **map layer**. This element is of type [MapLineRules](#).

Following is the parent element of the **MapLineLayer.MapLineRules** element.

Parent elements
<a href="#">MapLineLayer</a>

The following is the XML Schema definition of the **MapLineLayer.MapLineRules** element.

```
<xsd:element name="MapLineRules" type="MapLineRulesType" minOccurs="0" />
```

## 2.220.8 MapLineLayer.MapLines

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineLayer.MapLines** element specifies a collection of embedded **map lines**. This element is of type [MapLines](#).

Following is the parent element of the **MapLineLayer.MapLines** element.

Parent elements
<a href="#">MapLineLayer</a>

The following is the XML Schema definition of the **MapLineLayer.MapLines** element.

```
<xsd:element name="MapLines" type="MapLinesType" minOccurs="0" />
```

## 2.220.9 MapLineLayer.MapLineTemplate

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineLayer.MapLineTemplate** element specifies a **line template** to be applied for **lines** in the parent **line layer**. This element is optional. This element is of type [MapLineTemplate](#).

Following is the parent element of the **MapLineLayer.MapLineTemplate** element.

Parent elements
<a href="#">MapLineLayer</a>

The following is the XML Schema definition of the **MapLineLayer.MapLineTemplate** element.

```
<xsd:element name="MapLineTemplate" type="MapLineTemplateType" minOccurs="0" />
```

## 2.220.10 MapLineLayer.MapSpatialDataRegion

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineLayer.MapSpatialDataRegion** element specifies a data region as the source for the **map lines** for the parent layer. This element is optional. This element is ignored if the value of the **MapLineLayer.MapLines** element is specified.

The **MapLineLayer.MapSpatialDataRegion** cannot be present if any of the following elements is present:

- [MapLineLayer.MapSpatialDataSet](#)
- [MapLineLayer.MapShapefile](#)

The **MapLineLayer.MapSpatialDataRegion** element is of type [MapSpatialDataRegion](#).

Following is the parent element of the **MapLineLayer.MapSpatialDataRegion** element.

Parent elements
<a href="#">MapLineLayer</a>

The following is the XML Schema definition of the **MapLineLayer.MapSpatialDataRegion** element.

```
<xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType" minOccurs="0" />
```

### 2.220.11 MapLineLayer.MapShapefile

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLineLayer.MapShapefile** element specifies a **shapefile** as the source for **map lines** for the parent layer. This element is optional. This element is ignored if the value of the **MapLineLayer.MapLines** element is specified.

The **MapLineLayer.MapShapefile** element cannot be present if either of the following elements is present:

- **MapLineLayer.MapSpatialDataSet**
- **MapLineLayer.MapSpatialDataRegion**

The **MapLineLayer.MapShapefile** element is of type [MapShapefile](#).

Following is the parent element of the **MapLineLayer.MapShapefile** element.

Parent elements
<a href="#">MapLineLayer</a>

The following is the XML Schema definition of the **MapLineLayer.MapShapefile** element.

```
<xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
```

### 2.220.12 MapLineLayer.MapSpatialDataSet

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLineLayer.MapSpatialDataSet** element specifies a SQL Server dataset as the source for **map lines** for the parent layer. This element is optional. This element is ignored if [MapLineLayer.MapLines](#) is specified.

The **MapLineLayer.MapSpatialDataSet** element cannot be present if either of the following elements is present:

- [MapLineLayer.MapSpatialDataRegion](#)
- [MapLineLayer.MapShapefile](#)

The **MapLineLayer.MapSpatialDataSet** element is of type [MapSpatialDataSet](#).

Following is the parent element of the **MapLineLayer.MapSpatialDataSet** element.

Parent elements
<a href="#">MapLineLayer</a>

The following is the XML Schema definition of the **MapLineLayer.MapSpatialDataSet** element.



```
<xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType" minOccurs="0" />
```

### 2.220.13 MapLineLayer.MaximumZoom

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineLayer.MaximumZoom** element specifies the maximum zoom level at which the parent layer is visible. This element is optional.

If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 200. This element is ignored if the value of the **MapLineLayer.VisibilityMode** element is not "ZoomBased".

Following is the parent element of the **MapLineLayer.MaximumZoom** element.

Parent elements
<a href="#">MapLineLayer</a>

The following is the XML Schema definition of the **MapLineLayer.MaximumZoom** element.

```
<xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
```

### 2.220.14 MapLineLayer.MinimumZoom

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineLayer.MinimumZoom** element specifies the minimum zoom level at which the parent layer is visible. This element is optional.

If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 50. This element is ignored if the value of the **MapLineLayer.VisibilityMode** element is not "ZoomBased".

Following is the parent element of the **MapLineLayer.MinimumZoom** element.

Parent elements
<a href="#">MapLineLayer</a>

The following is the XML Schema definition of the **MapLineLayer.MinimumZoom** element.

```
<xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
```

### 2.220.15 MapLineLayer.Transparency

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineLayer.Transparency** element specifies the transparency for **map lines** in the parent layer as a percentage. This element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 100.

Following is the parent element of the **MapLineLayer.Transparency** element.

<b>Parent elements</b>
------------------------

<a href="#">MapLineLayer</a>
------------------------------

The following is the XML Schema definition of the **MapLineLayer.Transparency** element.

```
<xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
```

## 2.220.16 MapLineLayer.VisibilityMode

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLineLayer.VisibilityMode** element specifies the visibility mode for a map **line layer**. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Visible:** The layer is always visible.

**Hidden:** The layer is always hidden.

**ZoomBased:** The layer is shown in a zoom level range defined by [MapLineLayer.MaximumZoom](#) and [MapLineLayer.MinimumZoom](#).

If this element is not present, its value is interpreted as "Visible".

Following is the parent element of the **MapLineLayer.VisibilityMode** element.

<b>Parent elements</b>
------------------------

<a href="#">MapLineLayer</a>
------------------------------

The following is the XML Schema definition of the **MapLineLayer.VisibilityMode** element.

```
<xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
```

## 2.221 MapBindingFieldPairs

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapBindingFieldPairs** element specifies a collection of [MapBindingFieldPair](#) elements to be used to bind the **map items** in a **map polygon layer**, a map **point layer**, or a map **line layer** to the data region that is associated with the layer.

The **MapBindingFieldPairs** element MUST contain at least one **MapBindingFieldPair** instance.

The following are the parent elements and child element of the **MapBindingFieldPairs** element.

<b>Parent elements</b>
------------------------

<a href="#">MapLineLayer</a>
------------------------------

<a href="#">MapPointLayer</a>
-------------------------------

<b>Parent elements</b>
<a href="#">MapPolygonLayer</a>
<a href="#">MapElementView</a>

<b>Child elements</b>
<a href="#">MapBindingFieldPairs.MapBindingFieldPair</a>

The following is the XML Schema definition of the **MapBindingFieldPairs** element.

```
<xsd:complexType name="MapBindingFieldPairsType">
  <xsd:sequence>
    <xsd:element name="MapBindingFieldPair" type="MapBindingFieldPairType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.221.1 MapBindingFieldPairs.MapBindingFieldPair

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapBindingFieldPairs.MapBindingFieldPair** element specifies a field and an expression to be used to bind the **map items** in a [MapPolygonLayer](#), a [MapPointLayer](#), or a [MapLineLayer](#) to the data region that is associated with the layer. If the [MapBindingFieldPairs](#) element is a child of [MapElementView](#), the **MapBindingFieldPairs.MapBindingFieldPair** element is used to identify the map item to center on the viewport.

The **MapBindingFieldPairs.MapBindingFieldPair** element MUST be specified. This element is of type [MapBindingFieldPair](#).

Following is the parent element of the **MapBindingFieldPairs.MapBindingFieldPair** element.

<b>Parent elements</b>
<a href="#">MapBindingFieldPairs</a>

The following is the XML Schema definition of the **MapBindingFieldPairs.MapBindingFieldPair** element.

```
<xsd:element name="MapBindingFieldPair" type="MapBindingFieldPairType"
  minOccurs="1" maxOccurs="unbounded" />
```

### 2.222 MapBindingFieldPair

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapBindingFieldPair** element specifies a field and an expression to be used to bind the **spatial elements** in a [MapPolygonLayer](#), a [MapPointLayer](#), or a [MapLineLayer](#) to the data region that is associated with the layer.

If the [MapBindingFieldPairs](#) element is a child of [MapView](#), the [MapBindingFieldPairs.MapBindingFieldPair](#) element is used to identify the **map item** to center on the viewport.

The following are the parent and child elements of the **MapBindingFieldPair** element.

Parent elements
<a href="#">MapBindingFieldPairs</a>

Child elements
<a href="#">MapBindingFieldPair.BindingExpression</a>
<a href="#">MapBindingFieldPair.FieldName</a>

The following is the XML Schema definition of the **MapBindingFieldPair** element.

```
<xsd:complexType name="MapBindingFieldPairType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="FieldName" type="xsd:string" minOccurs="1" />
    <xsd:element name="BindingExpression" type="xsd:string" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.222.1 **MapBindingFieldPair.BindingExpression**

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapBindingFieldPair.BindingExpression** element specifies the expression to be evaluated for each innermost group instance of the data region that is compared to [MapBindingFieldPair.FieldName](#). This comparison occurs in order to perform the binding between the **spatial elements** in a map **polygon layer**, a map **point layer**, or a map line layer to the data region that is associated with the layer.

The **MapBindingFieldPair.BindingExpression** element MUST be specified.

Following is the parent element of the **MapBindingFieldPair.BindingExpression** element.

Parent elements
<a href="#">MapBindingFieldPair</a>

The following is the XML Schema definition of the **MapBindingFieldPair.BindingExpression** element.

```
<xsd:element name="BindingExpression" type="xsd:string" minOccurs="1" />
```

### 2.222.2 **MapBindingFieldPair.FieldName**

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapBindingFieldPair.FieldName** element specifies the field name of the **spatial element** in a map **polygon layer**, a map **point layer**, or a map line layer that is to be used for binding with the data region that is associated with the layer.

The **MapBindingFieldPair.FieldName** element MUST be specified. The value of this element MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **MapBindingFieldPair.FieldName** element.

Parent elements
<a href="#">MapBindingFieldPair</a>

The following is the XML Schema definition of the **MapBindingFieldPair.FieldName** element.

```
<xsd:element name="FieldName" type="xsd:string" minOccurs="1" />
```

## 2.223 MapFieldDefinitions

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapFieldDefinitions** element specifies a collection of [MapFieldDefinition](#) elements in a [MapPolygonLayer](#), a [MapPointLayer](#), or a [MapLineLayer](#). The **MapFieldDefinitions** element MUST contain at least one **MapFieldDefinition** element.

The following are the parent elements and child element of the **MapFieldDefinitions** element.

Parent elements
MapLineLayer
MapPointLayer
MapPolygonLayer

Child elements
<a href="#">MapFieldDefinitions.MapFieldDefinition</a>

The following is the XML Schema definition of the **MapFieldDefinitions** element.

```
<xsd:complexType name="MapFieldDefinitionsType">
  <xsd:sequence>
    <xsd:element name="MapFieldDefinition" type="MapFieldDefinitionType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.223.1 MapFieldDefinitions.MapFieldDefinition

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapFieldDefinitions.MapFieldDefinition** element specifies the metadata for a [MapField](#) element. The **MapFieldDefinitions.MapFieldDefinition** element MUST be specified. This element is of type [MapFieldDefinition](#).

Following is the parent element of the **MapFieldDefinitions.MapFieldDefinition** element.

Parent elements
<a href="#">MapFieldDefinitions</a>

The following is the XML Schema definition of the **MapFieldDefinitions.MapFieldDefinition** element.

```
<xsd:element name="MapFieldDefinition" type="MapFieldDefinitionType" minOccurs="1"
maxOccurs="unbounded" />
```

## 2.224 MapFieldDefinition

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapFieldDefinition** element specifies the metadata for a [MapField](#).

The following are the parent and child elements of the **MapFieldDefinition** element.

Parent elements
<a href="#">MapFieldDefinitions</a>

Child elements
<a href="#">MapFieldDefinition.DataType</a>
<a href="#">MapFieldDefinition.Name</a>

The following is the XML Schema definition of the **MapFieldDefinition** element.

```
<xsd:complexType name="MapFieldDefinitionType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" minOccurs="1" />
    <xsd:element name="DataType" minOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Boolean" />
          <xsd:enumeration value="DateTime" />
          <xsd:enumeration value="Integer" />
          <xsd:enumeration value="Float" />
          <xsd:enumeration value="String" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.224.1 MapFieldDefinition.DataType

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapFieldDefinition.DataType** element specifies the field data type. This element MUST be specified, and its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is one of the following:

**Boolean:** Specifies that the field value is a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2).

**DateTime:** Specifies that the field value is a [DateTime](#) ([XMLSCHEMA2] section 3.2.7).

**Integer:** Specifies that the field value is an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17).

**Float:** Specifies that the field value is a [Float](#) ([XMLSCHEMA2] section 3.2.4).

**String:** Specifies that the field value is a **String** ([XMLSCHEMA2/2] section 3.2.1).

Following is the parent element of the **MapFieldDefinition.DataType** element.

Parent elements
<a href="#">MapFieldDefinition</a>

The following is the XML Schema definition of the **MapFieldDefinition.DataType** element.

```
<xsd:element name="DataType" minOccurs="1">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Boolean" />
      <xsd:enumeration value="DateTime" />
      <xsd:enumeration value="Integer" />
      <xsd:enumeration value="Float" />
      <xsd:enumeration value="String" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.224.2 MapFieldDefinition.Name

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapFieldDefinition.Name** element specifies the name for a [MapFieldDefinition](#). The **MapFieldDefinition.Name** element MUST be specified, and its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1).

Following is the parent element of the **MapFieldDefinition.Name** element.

Parent elements
MapFieldDefinition

The following is the XML Schema definition of the **MapFieldDefinition.Name** element.

```
<xsd:element name="Name" type="xsd:string" minOccurs="1" />
```

## 2.225 MapLineRules

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineRules** element specifies a set of rules to be applied to **map lines** in the parent layer.

The following are the parent and child elements of the **MapLineRules** element.

Parent elements
<a href="#">MapLineLayer</a>

Child elements
<a href="#">MapLineRules.MapColorPaletteRule</a>
<a href="#">MapLineRules.MapColorRangeRule</a>
<a href="#">MapLineRules.MapCustomColorRule</a>
<a href="#">MapLineRules.MapSizeRule</a>

The following is the XML Schema definition of the **MapLineRules** element.

```
<xsd:complexType name="MapLineRulesType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType"
      minOccurs="0" />
    <xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType"
      minOccurs="0" />
    <xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType"
      minOccurs="0" />
    <xsd:element name="MapSizeRule" type="MapSizeRuleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.225.1 MapLineRules.MapColorPaletteRule

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineRules.MapColorPaletteRule** element specifies a palette of colors for map lines in the parent layer based on the value of the [MapColorPaletteRule.DataValue](#) element. The **MapLineRules.MapColorPaletteRule** element is optional. If this element is present, it overrides the [Style.Color](#) child element of the [MapLineTemplate](#) element.

The **MapLineRules.MapColorPaletteRule** element cannot be present if either of the following elements is present:

- [MapLineRules.MapColorRangeRule](#)
- [MapLineRules.MapCustomColorRule](#)

The **MapLineRules.MapColorPaletteRule** element is of type [MapColorPaletteRule](#).

Following is the parent element of the **MapLineRules.MapColorPaletteRule** element.



<b>Parent elements</b>
------------------------

<a href="#">MapLineRules</a>
------------------------------

The following is the XML Schema definition of the **MapLineRules.MapColorPaletteRule** element.

```
<xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType"
  minOccurs="0" />
```

## 2.225.2 MapLineRules.MapColorRangeRule

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapLineRules.MapColorRangeRule** element specifies a color range to be used for coloring map lines in the parent layer based on the value of the [MapColorRangeRule.DataValue](#) element. The **MapLineRules.MapColorRangeRule** element is optional. If this element is present, it overrides the [Style.Color](#) child element of the [MapLineTemplate](#) element.

The **MapLineRules.MapColorRangeRule** element cannot be present if either of the following elements is present:

- [MapLineRules.MapColorPaletteRule](#)
- [MapLineRules.MapCustomColorRule](#)

The **MapLineRules.MapColorRangeRule** element is of type [MapColorRangeRule](#).

Following is the parent element of the **MapLineRules.MapColorRangeRule** element.

<b>Parent elements</b>
------------------------

<a href="#">MapLineRules</a>
------------------------------

The following is the XML Schema definition of the **MapLineRules.MapColorRangeRule** element.

```
<xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType" minOccurs="0" />
```

## 2.225.3 MapLineRules.MapCustomColorRule

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapLineRules.MapCustomColorRule** element specifies a user-defined set of colors for map lines in the parent layer based on the value of the [MapCustomColorRule.DataValue](#) element. The **MapLineRules.MapCustomColorRule** element is optional. If this element is present, it overrides the [Style.Color](#) child element of the [MapLineTemplate](#) element.

The **MapLineRules.MapCustomColorRule** element cannot be present if either of the following elements is present:

- [MapLineRules.MapColorPaletteRule](#)
- [MapLineRules.MapColorRangeRule](#)

The **MapLineRules.MapCustomColorRule** element is of type [MapCustomColorRule](#).

Following is the parent element of the **MapLineRules.MapCustomColorRule** element.

<b>Parent elements</b>
<a href="#">MapLineRules</a>

The following is the XML Schema definition of the **MapLineRules.MapCustomColorRule** element.

```
<xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType" minOccurs="0" />
```

## 2.225.4 MapLineRules.MapSizeRule

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLineRules.MapSizeRule** element specifies a size range to be used for sizing map lines in the parent layer based on the value of the [MapSizeRule.DataValue](#) element. The **MapLineRules.MapSizeRule** element is optional.

If this element is present, it overrides the [MapLineTemplate.Width](#) element. This element is of type [MapSizeRule](#).

Following is the parent element of the **MapLineRules.MapSizeRule** element.

<b>Parent elements</b>
<a href="#">MapLineRules</a>

The following is the XML Schema definition of the **MapLineRules.MapSizeRule** element.

```
<xsd:element name="MapSizeRule" type="MapSizeRuleType" minOccurs="0" />
```

## 2.226 MapColorPaletteRule

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorPaletteRule** element specifies a palette of colors for **map items** in a [MapPolygonLayer](#), a [MapPointLayer](#), or a [MapLineLayer](#) based on the value of the [MapColorPaletteRule.DataValue](#) element.

The following are the parent and child elements of the **MapColorPaletteRule** element.

<b>Parent elements</b>
<a href="#">MapLineRules</a>
<a href="#">MapPointRules</a>
<a href="#">MapPolygonRules</a>

<b>Child elements</b>
<a href="#">MapColorPaletteRule.BucketCount</a>
<a href="#">MapColorPaletteRule.DataElementName</a>
<a href="#">MapColorPaletteRule.DataElementOutput</a>

Child elements
MapColorPaletteRule.DataValue
<a href="#">MapColorPaletteRule.DistributionType</a>
<a href="#">MapColorPaletteRule.EndValue</a>
<a href="#">MapColorPaletteRule.LegendName</a>
<a href="#">MapColorPaletteRule.LegendText</a>
<a href="#">MapColorPaletteRule.MapBuckets</a>
<a href="#">MapColorPaletteRule.Palette</a>
<a href="#">MapColorPaletteRule.ShowInColorScale</a>
<a href="#">MapColorPaletteRule.StartValue</a>

The following is the XML Schema definition of the **MapColorPaletteRule** element.

```
<xsd:complexType name="MapColorPaletteRuleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapAppearanceRule Start-->
    <xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapAppearanceRule End-->
    <!--MapColorRule Start-->
    <xsd:element name="ShowInColorScale" type="xsd:string" minOccurs="0" />
    <!--MapColorRule End-->
    <xsd:element name="Palette" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.226.1 MapColorPaletteRule.BucketCount

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapColorPaletteRule.BucketCount** element specifies the number of buckets for a [MapColorPaletteRule](#). The **MapColorPaletteRule.BucketCount** element is optional.

If the **MapColorPaletteRule.BucketCount** element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 5.

The **MapColorPaletteRule.BucketCount** element is ignored if the value of the [MapColorPaletteRule.DistributionType](#) element is "Custom" or if the value of the [MapColorPaletteRule.DataValue](#) element is not scalar.

Following is the parent element of the **MapColorPaletteRule.BucketCount** element.

Parent elements
<a href="#">MapColorPaletteRule</a>

The following is the XML Schema definition of the **MapColorPaletteRule.BucketCount** element.

```
<xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
```

## 2.226.2 MapColorPaletteRule.DataElementName

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorPaletteRule.DataElementName** element specifies the name to use for the data element or attribute for the [MapColorPaletteRule.DataValue](#) element. The **MapColorPaletteRule.DataElementName** element is optional.

If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is a **CLS-compliant identifier** [UTR15].

Following is the parent element of the **MapColorPaletteRule.DataElementName** element.

Parent elements
<a href="#">MapColorPaletteRule</a>

The following is the XML Schema definition of the **MapColorPaletteRule.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

## 2.226.3 MapColorPaletteRule.DataElementOutput

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorPaletteRule.DataElementOutput** element specifies whether the parent [MapColorPaletteRule](#) element will appear in a **data rendering**. The **MapColorPaletteRule.DataElementOutput** element is optional.

If this element is present, its value MUST be one of the following:

**Output:** Specifies that the item appears in the data rendering output.

**NoOutput:** Specifies that the item does not appear in the data rendering output.

If this element is not present, its value is interpreted as "Output".

Following is the parent element of the **MapColorPaletteRule.DataElementOutput** element.

Parent elements
<a href="#">MapColorPaletteRule</a>

The following is the XML Schema definition of the **MapColorPaletteRule.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.226.4 MapColorPaletteRule.DataValue

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorPaletteRule.DataValue** element specifies an expression to be evaluated on the scope of a map item in a [MapPolygonLayer](#), a [MapPointLayer](#), or a [MapLineLayer](#). The **MapColorPaletteRule.DataValue** element is optional.

Palette colors are assigned to map items based on the value of the **MapColorPaletteRule.DataValue** element. If this element is not present, each map item is assigned a distinct color from the palette.

Following is the parent element of the **MapColorPaletteRule.DataValue** element.

Parent elements
<a href="#">MapColorPaletteRule</a>

The following is the XML Schema definition of the **MapColorPaletteRule.DataValue** element.

```
<xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
```

## 2.226.5 MapColorPaletteRule.DistributionType

The **MapColorPaletteRule.DistributionType** element specifies the distribution type for a [MapColorPaletteRule](#) instance. The **MapColorPaletteRule.DistributionType** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Optimal:** The optimal distribution is applied.

**EqualInterval:** The equal-interval distribution is applied.

**EqualDistribution:** The equal distribution is applied.

**Custom:** The user-defined distribution is applied.

If this element is not present, its value is interpreted as "Optimal".

Following is the parent element of the **MapColorPaletteRule.DistributionType** element.

Parent elements
<a href="#">MapColorPaletteRule</a>

The following is the XML Schema definition of the **MapColorPaletteRule.DistributionType** element.

```
<xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
```

## 2.226.6 **MapColorPaletteRule.EndValue**

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorPaletteRule.EndValue** element specifies the end value for a [MapColorPaletteRule](#) distribution. The **MapColorPaletteRule.EndValue** element is optional.

The **MapColorPaletteRule.EndValue** element is ignored if the value of the [MapColorPaletteRule.DistributionType](#) element is "Custom" or if the value of the [MapColorPaletteRule.DataValue](#) element is not scalar. Otherwise, if the **MapColorPaletteRule.EndValue** element is not specified, its value MUST be automatically calculated.

Following is the parent element of the **MapColorPaletteRule.EndValue** element.

Parent elements
<a href="#">MapColorPaletteRule</a>

The following is the XML Schema definition of the **MapColorPaletteRule.EndValue** element.

```
<xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
```

## 2.226.7 **MapColorPaletteRule.LegendName**

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorPaletteRule.LegendName** specifies the name of the legend in which the rule distribution is displayed. The **MapColorPaletteRule.LegendName** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). If this element is not present, the rule distribution MUST NOT be displayed in any legend.

Following is the parent element of the **MapColorPaletteRule.LegendName** element.

Parent elements
<a href="#">MapColorPaletteRule</a>

The following is the XML Schema definition of the **MapColorPaletteRule.LegendName** element.

```
<xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
```

## 2.226.8 MapColorPaletteRule.LegendText

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorPaletteRule.LegendText** element specifies the label to be displayed for the map distribution in the legend that is specified by [MapColorPaletteRule.LegendName](#). The **MapColorPaletteRule.LegendText** element is optional.

If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as the start value of the bucket, followed by a hyphen (-), followed by the end value of the bucket.

Following is the parent element of the **MapColorPaletteRule.LegendText** element.

Parent elements
<a href="#">MapColorPaletteRule</a>

The following is the XML Schema definition of the **MapColorPaletteRule.LegendText** element.

```
<xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
```

## 2.226.9 MapColorPaletteRule.MapBuckets

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorPaletteRule.MapBuckets** element specifies a custom set of [MapBucket](#) instances to be used for data distribution. The **MapColorPaletteRule.MapBuckets** element is optional.

The **MapColorPaletteRule.MapBuckets** element MUST be specified if the value of the [MapColorPaletteRule.DistributionType](#) element is specified as "Custom". Otherwise, the **MapColorPaletteRule.MapBuckets** element is ignored. This element is of type [MapBuckets](#).

Following is the parent element of the **MapColorPaletteRule.MapBuckets** element.

Parent elements
<a href="#">MapColorPaletteRule</a>

The following is the XML Schema definition of the **MapColorPaletteRule.MapBuckets** element.

```
<xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
```

## 2.226.10 MapColorPaletteRule.Palette

The **MapColorPaletteRule.Palette** element specifies the palette to be used for coloring map items. This element is optional. If the **MapColorPaletteRule.Palette** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Random:** A random palette is used.

**Light:** A light palette is used.

**SemiTransparent:** A semitransparent palette is used.

**BrightPastel:** A bright pastel palette is used.

If this element is not present, its value is interpreted as "Random".

Following is the parent element of the **MapColorPaletteRule.Palette** element.

Parent elements
<a href="#">MapColorPaletteRule</a>

The following is the XML Schema definition of the **MapColorPaletteRule.Palette** element.

```
<xsd:element name="Palette" type="xsd:string" minOccurs="0" />
```

## 2.226.11 MapColorPaletteRule.ShowInColorScale

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorPaletteRule.ShowInColorScale** element specifies whether the colors that are assigned to map items are displayed in a [MapColorScale](#) instance. The **MapColorPaletteRule.ShowInColorScale** element is optional.

If the **MapColorPaletteRule.ShowInColorScale** element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapColorPaletteRule.ShowInColorScale** element.

Parent elements
<a href="#">MapColorPaletteRule</a>

The following is the XML Schema definition of the **MapColorPaletteRule.ShowInColorScale** element.

```
<xsd:element name="ShowInColorScale" type="xsd:string" minOccurs="0" />
```

## 2.226.12 MapColorPaletteRule.StartValue

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorPaletteRule.StartValue** specifies the start value for a [MapColorPaletteRule](#) distribution. The **MapColorPaletteRule.StartValue** element is optional.

This element is ignored if the value of the [MapColorPaletteRule.DistributionType](#) element is "Custom" or if the value of the [MapColorPaletteRule.DataValue](#) element is not scalar. Otherwise, if the **MapColorPaletteRule.StartValue** element is not specified, its value MUST be automatically calculated.

Following is the parent element of the **MapColorPaletteRule.StartValue** element.

Parent elements
<a href="#">MapColorPaletteRule</a>



The following is the XML Schema definition of the **MapColorPaletteRule.StartValue** element.

```
<xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
```

## 2.227 MapBuckets

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapBuckets** element specifies a collection of [MapBucket](#) elements to be used for analytical data distribution in a [MapColorRangeRule](#), a [MapColorPaletteRule](#), a [MapCustomColorRule](#), a [MapSizeRule](#), or a [MapMarkerRule](#).

The **MapBuckets** element is specified if the value of the [MapColorRangeRule.DistributionType](#), [MapColorPaletteRule.DistributionType](#), [MapCustomColorRule.DistributionType](#), [MapSizeRule.DistributionType](#), or [MapMarkerRule.DistributionType](#) element is specified as "Custom". Otherwise, the **MapBuckets** element is ignored. If this element is present, it MUST contain at least one **MapBucket** instance.

The following are the parent elements and child element of the **MapBuckets** element.

Parent elements
<a href="#">MapColorPaletteRule</a>
<a href="#">MapColorRangeRule</a>
<a href="#">MapCustomColorRule</a>
<a href="#">MapMarkerRule</a>
<a href="#">MapSizeRule</a>

Child elements
<a href="#">MapBuckets.MapBucket</a>

The following is the XML Schema definition of the **MapBuckets** element.

```
<xsd:complexType name="MapBucketsType">
  <xsd:sequence>
    <xsd:element name="MapBucket" type="MapBucketType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.227.1 MapBuckets.MapBucket

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapBuckets.MapBucket** element specifies the range of the custom start and end values to be used for analytical data distribution in a [MapColorRangeRule](#), a [MapColorPaletteRule](#), a [MapCustomColorRule](#), a [MapSizeRule](#), or a [MapMarkerRule](#).

The **MapBuckets.MapBucket** element MUST be specified. This element is of type [MapBucket](#).

Following is the parent element of the **MapBuckets.MapBucket** element.

Parent elements
<a href="#">MapBuckets</a>

The following is the XML Schema definition of the **MapBuckets.MapBucket** element.

```
<xsd:element name="MapBucket" type="MapBucketType" minOccurs="1"
maxOccurs="unbounded" />
```

## 2.228 MapBucket

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapBucket** element specifies the range of custom start and end values to be used for analytical data distribution in a [MapColorRangeRule](#), a [MapColorPaletteRule](#), a [MapCustomColorRule](#), a [MapSizeRule](#), or a [MapMarkerRule](#).

The following are the parent and child elements of the **MapBucket** element.

Parent elements
<a href="#">MapBuckets</a>

Child elements
<a href="#">MapBucket.EndValue</a>
<a href="#">MapBucket.StartValue</a>

The following is the XML Schema definition of the **MapBucket** element.

```
<xsd:complexType name="MapBucketType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.228.1 MapBucket.EndValue

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapBucket.EndValue** element specifies the end value for a **map bucket**. This element is optional. If this element is not present, its value is interpreted as automatically calculated.

Following is the parent element of the **MapBucket.EndValue** element.

Parent elements
<a href="#">MapBucket</a>

The following is the XML Schema definition of the **MapBucket.EndValue** element.

```
<xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
```

## 2.228.2 MapBucket.StartValue

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapBucket.StartValue** element specifies the start value for a [MapBucket](#). This element is optional. If this element is not present, its value is interpreted as automatically calculated.

Following is the parent element of the **MapBucket.StartValue** element.

Parent elements
MapBucket

The following is the XML Schema definition of the **MapBucket.StartValue** element.

```
<xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
```

## 2.229 MapColorRangeRule

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorRangeRule** element specifies a color range to be used for coloring **map items** in a [MapPolygonLayer](#), a [MapPointLayer](#), or a [MapLineLayer](#), based on the value of the [MapColorRangeRule.DataValue](#) element.

The following are the parent and child elements of the **MapColorRangeRule** element.

Parent elements
<a href="#">MapLineRules</a>
<a href="#">MapPointRules</a>
<a href="#">MapPolygonRules</a>

Child elements
<a href="#">MapColorRangeRule.BucketCount</a>
<a href="#">MapColorRangeRule.DataElementName</a>
<a href="#">MapColorRangeRule.DataElementOutput</a>
MapColorRangeRule.DataValue

Child elements
<a href="#">MapColorRangeRule.DistributionType</a>
<a href="#">MapColorRangeRule.EndColor</a>
<a href="#">MapColorRangeRule.EndValue</a>
<a href="#">MapColorRangeRule.LegendName</a>
<a href="#">MapColorRangeRule.LegendText</a>
<a href="#">MapColorRangeRule.MapBuckets</a>
<a href="#">MapColorRangeRule.MiddleColor</a>
<a href="#">MapColorRangeRule.ShowInColorScale</a>
<a href="#">MapColorRangeRule.StartColor</a>
<a href="#">MapColorRangeRule.StartValue</a>

The following is the XML Schema definition of the **MapColorRangeRule** element.

```
<xsd:complexType name="MapColorRangeRuleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapAppearanceRule Start-->
    <xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapAppearanceRule End-->
    <!--MapColorRule Start-->
    <xsd:element name="ShowInColorScale" type="xsd:string" minOccurs="0" />
    <!--MapColorRule End-->
    <xsd:element name="StartColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="MiddleColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.229.1 MapColorRangeRule.BucketCount

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorRangeRule.BucketCount** element specifies the number of buckets for a [MapColorRangeRule](#). The **MapColorRangeRule.BucketCount** element is optional.

If the **MapColorRangeRule.BucketCount** element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 5. This element is ignored if the value of the [MapColorRangeRule.DistributionType](#) element is "Custom" or if the value of the [MapColorRangeRule.DataValue](#) element is not scalar.

Following is the parent element of the **MapColorRangeRule.BucketCount** element.

Parent elements
<a href="#">MapColorRangeRule</a>

The following is the XML Schema definition of the **MapColorRangeRule.BucketCount** element.

```
<xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
```

## 2.229.2 MapColorRangeRule.DataElementName

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorRangeRule.DataElementName** element specifies the name to use for a data element or attribute for the [MapColorRangeRule.DataValue](#) element. The **MapColorRangeRule.DataElementName** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is a **CLS-compliant identifier** [UTR15].

Following is the parent element of the **MapColorRangeRule.DataElementName** element.

Parent elements
<a href="#">MapColorRangeRule</a>

The following is the XML Schema definition of the **MapColorRangeRule.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

## 2.229.3 MapColorRangeRule.DataElementOutput

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorRangeRule.DataElementOutput** element indicates whether the parent [MapColorRangeRule](#) element will appear in a **data rendering**. The **MapColorRangeRule.DataElementOutput** element is optional.

If this element is present, its value MUST be one of the following:

**Output:** Specifies that the item appears in the data rendering output.

**NoOutput:** Specifies that the item does not appear in the data rendering output.

If this element is not present, its value is interpreted as "Output".

Following is the parent element of the **MapColorRangeRule.DataElementOutput** element.

<b>Parent elements</b>
------------------------

<a href="#">MapColorRangeRule</a>
-----------------------------------

The following is the XML Schema definition of the **MapColorRangeRule.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.229.4 MapColorRangeRule.DataValue

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorRangeRule.DataValue** element specifies an expression to be evaluated on the scope of a [Map](#) element in a [MapPolygonLayer](#), a [MapPointLayer](#), or a [MapLineLayer](#). The **MapColorRangeRule.DataValue** element is optional.

If this element is present, the color range that is specified in the element is assigned to the map item based on this element. If the **MapColorRangeRule.DataValue** element is not present, each map item is assigned a distinct color from the color range specified.

Following is the parent element of the **MapColorRangeRule.DataValue** element.

<b>Parent elements</b>
------------------------

<a href="#">MapColorRangeRule</a>
-----------------------------------

The following is the XML Schema definition of the **MapColorRangeRule.DataValue** element.

```
<xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
```

## 2.229.5 MapColorRangeRule.DistributionType

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorRangeRule.DistributionType** element specifies the distribution type for a [MapColorRangeRule](#) instance. The **MapColorRangeRule.DistributionType** element is optional. If the **MapColorRangeRule.DistributionType** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Optimal:** The optimal distribution is applied.

**EqualInterval:** The equal-interval distribution is applied.

**EqualDistribution:** The equal distribution is applied.

**Custom:** The user-defined distribution is applied.

If this element is not present, its value is interpreted as "Optimal".

Following is the parent element of the **MapColorRangeRule.DistributionType** element.

Parent elements
MapColorRangeRule

The following is the XML Schema definition of the **MapColorRangeRule.DistributionType** element.

```
<xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
```

## 2.229.6 MapColorRangeRule.EndColor

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorRangeRule.EndColor** element specifies the end color for a [MapColorRangeRule](#) distribution. The **MapColorRangeRule.EndColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. If this element is not present, its value is interpreted as "Red".

Following is the parent element of the **MapColorRangeRule.EndColor** element.

Parent elements
MapColorRangeRule

The following is the XML Schema definition of the **MapColorRangeRule.EndColor** element.

```
<xsd:element name="EndColor" type="xsd:string" minOccurs="0" />
```

## 2.229.7 MapColorRangeRule.EndValue

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorRangeRule.EndValue** specifies the end value for a [MapColorRangeRule](#) distribution. The **MapColorRangeRule.EndValue** element is optional.

This element is ignored if the value of the [MapColorRangeRule.DistributionType](#) element is "Custom" or if the value of the [MapColorRangeRule.DataValue](#) element is not scalar. Otherwise, if the **MapColorRangeRule.EndValue** element is not present, its value MUST be automatically calculated.

Following is the parent element of the **MapColorRangeRule.EndValue** element.

Parent elements
MapColorRangeRule

The following is the XML Schema definition of the **MapColorRangeRule.EndValue** element.

```
<xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
```

## 2.229.8 **MapColorRangeRule.LegendName**

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorRangeRule.LegendName** element specifies the name of the legend instance in which a rule distribution is displayed. The **MapColorRangeRule.LegendName** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). If this element is not present, the rule distribution MUST NOT be displayed in any legend.

Following is the parent element of the **MapColorRangeRule.LegendName** element.

Parent elements
<a href="#">MapColorRangeRule</a>

The following is the XML Schema definition of the **MapColorRangeRule.LegendName** element.

```
<xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
```

## 2.229.9 **MapColorRangeRule.LegendText**

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorRangeRule.LegendText** element specifies the label to display for the map distribution in the legend instance that is specified by the [MapColorRangeRule.LegendName](#) element. The **MapColorRangeRule.LegendText** element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as the start value of the bucket, followed by a hyphen (-), followed by the end value of the bucket.

Following is the parent element of the **MapColorRangeRule.LegendText** element.

Parent elements
<a href="#">MapColorRangeRule</a>

The following is the XML Schema definition of the **MapColorRangeRule.LegendText** element.

```
<xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
```

## 2.229.10 **MapColorRangeRule.MapBuckets**

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapColorRangeRule.MapBuckets** element specifies a custom set of [MapBucket](#) instances to be used for data distribution. The **MapColorRangeRule.MapBuckets** element is optional.

This element MUST be specified if the value of the [MapColorRangeRule.DistributionType](#) element is "Custom". Otherwise, the **MapColorRangeRule.MapBuckets** element is ignored. This element is of type [MapBuckets](#).

Following is the parent element of the **MapColorRangeRule.MapBuckets** element.



<b>Parent elements</b>
------------------------

<a href="#">MapColorRangeRule</a>
-----------------------------------

The following is the XML Schema definition of the **MapColorRangeRule.MapBuckets** element.

```
<xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
```

### 2.229.11 **MapColorRangeRule.MiddleColor**

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapColorRangeRule.MiddleColor** element specifies the middle color for a [MapColorRangeRule](#) distribution. The **MapColorRangeRule.MiddleColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. If this element is not present, its value is interpreted as "Yellow".

Following is the parent element of the **MapColorRangeRule.MiddleColor** element.

<b>Parent elements</b>
------------------------

<a href="#">MapColorRangeRule</a>
-----------------------------------

The following is the XML Schema definition of the **MapColorRangeRule.MiddleColor** element.

```
<xsd:element name="MiddleColor" type="xsd:string" minOccurs="0" />
```

### 2.229.12 **MapColorRangeRule.ShowInColorScale**

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapColorRangeRule.ShowInColorScale** element specifies whether the colors assigned to map items are displayed in a [MapColorScale](#) instance. The **MapColorRangeRule.ShowInColorScale** element is optional. If this element is present, its value MUST be a **Boolean** ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapColorRangeRule.ShowInColorScale** element.

<b>Parent elements</b>
------------------------

<a href="#">MapColorRangeRule</a>
-----------------------------------

The following is the XML Schema definition of the **MapColorRangeRule.ShowInColorScale** element.

```
<xsd:element name="ShowInColorScale" type="xsd:string" minOccurs="0" />
```

### 2.229.13 **MapColorRangeRule.StartColor**

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapColorRangeRule.StartColor** element specifies the start color for a [MapColorRangeRule](#) distribution. The **MapColorRangeRule.StartColor** element is optional. If this element is present, its

value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. If this element is not present, its value is interpreted as "Green".

Following is the parent element of the **MapColorRangeRule.StartColor** element.

Parent elements
<a href="#">MapColorRangeRule</a>

The following is the XML Schema definition of the **MapColorRangeRule.StartColor** element.

```
<xsd:element name="StartColor" type="xsd:string" minOccurs="0" />
```

## 2.229.14 MapColorRangeRule.StartValue

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapColorRangeRule.StartValue** element specifies the start value for a [MapColorRangeRule](#) distribution. The **MapColorRangeRule.StartValue** element is optional.

The **MapColorRangeRule.StartValue** element is ignored if the value of the [MapColorPaletteRule.DistributionType](#) element is "Custom" or if the value of the [MapColorPaletteRule.DataValue](#) element is not scalar. Otherwise, if the **MapColorRangeRule.StartValue** element is not present, its value MUST be automatically calculated.

Following is the parent element of the **MapColorRangeRule.StartValue** element.

Parent elements
<a href="#">MapColorRangeRule</a>

The following is the XML Schema definition of the **MapColorRangeRule.StartValue** element.

```
<xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
```

## 2.230 MapCustomColorRule

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapCustomColorRule** element specifies a user-defined set of colors for the **map items** in a [MapPolygonLayer](#), a [MapPointLayer](#), or a [MapLineLayer](#) based on the value of the [MapColorPaletteRule.DataValue](#) element.

The following are the parent and child elements of the **MapCustomColorRule** element.

Parent elements
<a href="#">MapLineRules</a>
<a href="#">MapPointRules</a>
<a href="#">MapPolygonRules</a>

Child elements
<a href="#">MapCustomColorRule.BucketCount</a>
<a href="#">MapCustomColorRule.DataElementName</a>
<a href="#">MapCustomColorRule.DataElementOutput</a>
<a href="#">MapCustomColorRule.DataValue</a>
<a href="#">MapCustomColorRule.DistributionType</a>
<a href="#">MapCustomColorRule.EndValue</a>
<a href="#">MapCustomColorRule.LegendName</a>
<a href="#">MapCustomColorRule.LegendText</a>
<a href="#">MapCustomColorRule.MapBuckets</a>
<a href="#">MapCustomColorRule.MapCustomColors</a>
<a href="#">MapCustomColorRule.ShowInColorScale</a>
<a href="#">MapCustomColorRule.StartValue</a>

The following is the XML Schema definition of the **MapCustomColorRule** element.

```

<xsd:complexType name="MapCustomColorRuleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapAppearanceRule Start-->
    <xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapAppearanceRule End-->
    <!--MapColorRule Start-->
    <xsd:element name="ShowInColorScale" type="xsd:string" minOccurs="0" />
    <!--MapColorRule End-->
    <xsd:element name="MapCustomColors" type="MapCustomColorsType"
      minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.230.1 MapCustomColorRule.BucketCount

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapCustomColorRule.BucketCount** element specifies the number of buckets for a [MapCustomColorRule](#) distribution. The **MapCustomColorRule.BucketCount** element is optional.

If the **MapCustomColorRule.BucketCount** element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 5. This element is ignored if the value of the [MapCustomColorRule.DistributionType](#) element is "Custom" or if the value of the [MapCustomColorRule.DataValue](#) element is not scalar.

Following is the parent element of the **MapCustomColorRule.BucketCount** element.

Parent elements
<a href="#">MapCustomColorRule</a>

The following is the XML Schema definition of the **MapCustomColorRule.BucketCount** element.

```
<xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
```

### 2.230.2 MapCustomColorRule.DataElementName

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapCustomColorRule.DataElementName** element specifies the name to use for the data element or attribute for the [MapCustomColorRule.DataValue](#) element. The **MapCustomColorRule.DataElementName** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is a **CLS-compliant identifier** [UTR15].

Following is the parent element of the **MapCustomColorRule.DataElementName** element.

Parent elements
<a href="#">MapCustomColorRule</a>

The following is the XML Schema definition of the **MapCustomColorRule.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.230.3 MapCustomColorRule.DataElementOutput

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapCustomColorRule.DataElementOutput** element indicates whether the parent [MapCustomColorRule](#) element will appear in a **data rendering**. The **MapCustomColorRule.DataElementOutput** element is optional.

If this element is present, its value MUST be one of the following:

**Output:** Specifies that the item appears in the data rendering output.

**NoOutput:** Specifies that the item does not appear in the data rendering output.

If this element is not present, its value is interpreted as "Output".

Following is the parent element of the **MapCustomColorRule.DataElementOutput** element.

<b>Parent elements</b>
------------------------

<a href="#">MapCustomColorRule</a>
------------------------------------

The following is the XML Schema definition of the **MapCustomColorRule.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

#### 2.230.4 MapCustomColorRule.DataValue

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapCustomColorRule.DataValue** element specifies an expression to be evaluated on the scope of the [Map](#) element in a [MapPolygonLayer](#), a [MapPointLayer](#), or a [MapLineLayer](#). The **MapCustomColorRule.DataValue** element is optional.

If the **MapCustomColorRule.DataValue** element is present, the specified custom colors are assigned to the **Map** elements based on this element. If this element is not present, each **Map** element is assigned a distinct custom color.

Following is the parent element of the **MapCustomColorRule.DataValue** element.

<b>Parent elements</b>
------------------------

<a href="#">MapCustomColorRule</a>
------------------------------------

The following is the XML Schema definition of the **MapCustomColorRule.DataValue** element.

```
<xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
```

#### 2.230.5 MapCustomColorRule.DistributionType

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapCustomColorRule.DistributionType** element specifies the distribution type for a [MapCustomColorRule](#) instance. The **MapCustomColorRule.DistributionType** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Optimal:** The optimal distribution is applied.

**EqualInterval:** The equal-interval distribution is applied.

**EqualDistribution:** The equal distribution is applied.

**Custom:** The user-defined distribution is applied.

If this element is not present, its value is interpreted as "Optimal".

Following is the parent element of the **MapCustomColorRule.DistributionType** element.

Parent elements
MapCustomColorRule

The following is the XML Schema definition of the **MapCustomColorRule.DistributionType** element.

```
<xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
```

### 2.230.6 MapCustomColorRule.EndValue

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapCustomColorRule.EndValue** element specifies the end value for a [MapCustomColorRule](#) distribution. The **MapCustomColorRule.EndValue** element is optional.

This element is ignored if the value of the [MapCustomColorRule.DistributionType](#) element is "Custom" or if the value of the [MapCustomColorRule.DataValue](#) element is not scalar. Otherwise, if the **MapCustomColorRule.EndValue** element is not present, its value MUST be automatically calculated.

Following is the parent element of the **MapCustomColorRule.EndValue** element.

Parent elements
MapCustomColorRule

The following is the XML Schema definition of the **MapCustomColorRule.EndValue** element.

```
<xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
```

### 2.230.7 MapCustomColorRule.LegendName

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapCustomColorRule.LegendName** element specifies the name of the legend instance in which a rule distribution is displayed. The **MapCustomColorRule.LegendName** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). If this element is not present, the rule distribution MUST NOT be displayed in any legend.

Following is the parent element of the **MapCustomColorRule.LegendName** element.

Parent elements
<a href="#">MapCustomColorRule</a>

The following is the XML Schema definition of the **MapCustomColorRule.LegendName** element.

```
<xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
```

## 2.230.8 MapCustomColorRule.LegendText

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapCustomColorRule.LegendText** element specifies the label to be displayed for the map distribution in the legend instance that is specified by the [MapCustomColorRule.LegendName](#) element. The **MapCustomColorRule.LegendText** element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as the start value of the bucket, followed by a hyphen (-), followed by the end value of the bucket.

Following is the parent element of the **MapCustomColorRule.LegendText** element.

Parent elements
<a href="#">MapCustomColorRule</a>

The following is the XML Schema definition of the **MapCustomColorRule.LegendText** element.

```
<xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
```

## 2.230.9 MapCustomColorRule.MapBuckets

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapCustomColorRule.MapBuckets** element specifies a custom set of [MapBucket](#) instances to be used for data distribution. The **MapCustomColorRule.MapBuckets** element is optional.

This element MUST be specified if the value of the [MapCustomColorRule.DistributionType](#) element is "Custom." Otherwise, the value of the **MapCustomColorRule.MapBuckets** element is ignored. This element is of type [MapBuckets](#).

Following is the parent element of the **MapCustomColorRule.MapBuckets** element.

Parent elements
<a href="#">MapCustomColorRule</a>

The following is the XML Schema definition of the **MapCustomColorRule.MapBuckets** element.

```
<xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
```

## 2.230.10 MapCustomColorRule.MapCustomColors

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapCustomColorRule.MapCustomColors** element specifies a collection of colors to be used to colorize map **spatial elements**. This element MUST be specified. This element is of type [MapCustomColors](#).

Following is the parent element of the **MapCustomColorRule.MapCustomColors** element.

<b>Parent elements</b>
------------------------

<a href="#">MapCustomColorRule</a>
------------------------------------

The following is the XML Schema definition of the **MapCustomColorRule.MapCustomColors** element.

```
<xsd:element name="MapCustomColors" type="MapCustomColorsType" minOccurs="1" />
```

### 2.230.11 MapCustomColorRule.ShowInColorScale

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapColorRangeRule.ShowInColorScale** element specifies whether the colors assigned to a map item will be displayed in the [MapColorScale](#) instance. The **MapColorRangeRule.ShowInColorScale** element is optional.

If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapColorRangeRule.ShowInColorScale** element.

<b>Parent elements</b>
------------------------

<a href="#">MapCustomColorRule</a>
------------------------------------

The following is the XML Schema definition of the **MapColorRangeRule.ShowInColorScale** element.

```
<xsd:element name="ShowInColorScale" type="xsd:string" minOccurs="0" />
```

### 2.230.12 MapCustomColorRule.StartValue

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapCustomColorRule.StartValue** element specifies the start value for a [MapCustomColorRule](#) distribution. The **MapCustomColorRule.StartValue** element is optional.

This element is ignored if the value of the [MapColorPaletteRule.DistributionType](#) element is "Custom" or if the value of the [MapColorPaletteRule.DataValue](#) element is not scalar. Otherwise, if the **MapCustomColorRule.StartValue** element is not present, its value MUST be automatically calculated.

Following is the parent element of the **MapCustomColorRule.StartValue** element.

<b>Parent elements</b>
------------------------

<a href="#">MapCustomColorRule</a>
------------------------------------

The following is the XML Schema definition of the **MapCustomColorRule.StartValue** element.

```
<xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
```



## 2.231 MapCustomColors

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapCustomColors** element specifies a collection of **MapCustomColor** elements to be used to colorize **map spatial elements**. The **MapCustomColors** element MUST contain at least one **MapCustomColor** element.

The following are the parent and child elements of the **MapCustomColors** element.

Parent elements
<a href="#">MapCustomColorRule</a>

Child elements
<a href="#">MapCustomColors.MapCustomColor</a>

The following is the XML Schema definition of the **MapCustomColors** element.

```
<xsd:complexType name="MapCustomColorsType">
  <xsd:sequence>
    <xsd:element name="MapCustomColor" type="xsd:string" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.231.1 MapCustomColors.MapCustomColor

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapCustomColors.MapCustomColor** element specifies a collection of **MapCustomColor** elements to be used to colorize map spatial elements. The **MapCustomColors.MapCustomColor** element MUST be specified. The value of this element MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

Following is the parent element of the **MapCustomColors.MapCustomColor** element.

Parent elements
<a href="#">MapCustomColors</a>

The following is the XML Schema definition of the **MapCustomColors.MapCustomColor** element.

```
<xsd:element name="MapCustomColor" type="xsd:string" minOccurs="1"
  maxOccurs="unbounded" />
```

## 2.232 MapSizeRule

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapSizeRule** element specifies a size range to use for sizing **map items** in a [MapPointLayer](#) or a [MapLineLayer](#). This size range is based on the value of the [MapSizeRule.DataValue](#) element.

The following are the parent and child elements of the **MapSizeRule** element.

Parent elements
<a href="#">MapLineRules</a>
<a href="#">MapPointRules</a>

Child elements
<a href="#">MapSizeRule.BucketCount</a>
<a href="#">MapSizeRule.DataElementName</a>
<a href="#">MapSizeRule.DataElementOutput</a>
MapSizeRule.DataValue
<a href="#">MapSizeRule.DistributionType</a>
<a href="#">MapSizeRule.EndSize</a>
<a href="#">MapSizeRule.EndValue</a>
<a href="#">MapSizeRule.LegendName</a>
<a href="#">MapSizeRule.LegendText</a>
<a href="#">MapSizeRule.MapBuckets</a>
<a href="#">MapSizeRule.StartSize</a>
<a href="#">MapSizeRule.StartValue</a>

The following is the XML Schema definition of the **MapSizeRule** element.

```
<xsd:complexType name="MapSizeRuleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapAppearanceRule Start-->
    <xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapAppearanceRule End-->
    <xsd:element name="StartSize" type="xsd:string" minOccurs="1" />
    <xsd:element name="EndSize" type="xsd:string" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.232.1 MapSizeRule.BucketCount

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSizeRule.BucketCount** element specifies the number of buckets for a [MapSizeRule](#). The **MapSizeRule.BucketCount** element is optional.

If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 5. This element is ignored if the value of the [MapSizeRule.DistributionType](#) element is "Custom" or if the value of the [MapSizeRule.DataValue](#) element is not scalar.

Following is the parent element of the **MapSizeRule.BucketCount** element.

Parent elements
<a href="#">MapSizeRule</a>

The following is the XML Schema definition of the **MapSizeRule.BucketCount** element.

```
<xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
```

### 2.232.2 MapSizeRule.DataElementName

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSizeRule.DataElementName** element specifies the name to use for the data element or attribute of the [MapSizeRule.DataValue](#) element. The **MapSizeRule.DataElementName** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#).

Following is the parent element of the **MapSizeRule.DataElementName** element.

Parent elements
<a href="#">MapSizeRule</a>

The following is the XML Schema definition of the **MapSizeRule.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.232.3 MapSizeRule.DataElementOutput

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSizeRule.DataElementOutput** element specifies whether the parent [MapSizeRule](#) element will appear in a **data rendering**. The **MapSizeRule.DataElementOutput** is optional.

If this element is present, its value MUST be one of the following:

**Output:** Specifies that the item appears in the data rendering output.

**NoOutput:** Specifies that the item does not appear in the data rendering output.

If this element is not present, its value is interpreted as "Output".

Following is the parent element of the **MapSizeRule.DataElementOutput** element.

Parent elements
<a href="#">MapSizeRule</a>

The following is the XML Schema definition of the **MapSizeRule.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

#### 2.232.4 **MapSizeRule.DataValue**

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapSizeRule.DataValue** element specifies an expression to be evaluated on the scope of the map item in a [MapPolygonLayer](#), a [MapPointLayer](#), or a [MapLineLayer](#). The **MapSizeRule.DataValue** element is optional.

The specified size range is assigned to the map items based on this element. If the **MapSizeRule.DataValue** element is not specified, each map item is assigned a distinct size in the size range.

Following is the parent element of the **MapSizeRule.DataValue** element.

Parent elements
<a href="#">MapSizeRule</a>

The following is the XML Schema definition of the **MapSizeRule.DataValue** element.

```
<xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
```

#### 2.232.5 **MapSizeRule.DistributionType**

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapSizeRule.DistributionType** element specifies the distribution type for a [MapSizeRule](#) instance. The **MapSizeRule.DistributionType** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Optimal:** The optimal distribution is applied.

**EqualInterval:** The equal-interval distribution is applied.

**EqualDistribution:** The equal distribution is applied.

**Custom:** The user-defined distribution is applied.

If this element is not present, its value is interpreted as "Optimal".

Following is the parent element of the **MapSizeRule.DistributionType** element.

Parent elements
MapSizeRule

The following is the XML Schema definition of the **MapSizeRule.DistributionType** element.

```
<xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
```

### 2.232.6 MapSizeRule.EndSize

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSizeRule.EndSize** element specifies the end size of the size range to be applied to **map points** or map lines. This end size is based on the value of the [MapSizeRule.DataValue](#) element.

Following is the parent element of the **MapSizeRule.EndSize** element.

Parent elements
<a href="#">MapSizeRule</a>

The following is the XML Schema definition of the **MapSizeRule.EndSize** element.

```
<xsd:element name="EndSize" type="xsd:string" minOccurs="1" />
```

### 2.232.7 MapSizeRule.EndValue

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSizeRule.EndValue** element specifies the end value for a [MapSizeRule](#) distribution. The **MapSizeRule.EndValue** element is optional.

This element is ignored if the value of the [MapSizeRule.DistributionType](#) element is "Custom" or if the value of the [MapSizeRule.DataValue](#) element is not scalar. Otherwise, if the **MapSizeRule.EndValue** element is not specified, its value MUST be automatically calculated.

Following is the parent element of the **MapSizeRule.EndValue** element.

Parent elements
MapSizeRule

The following is the XML Schema definition of the **MapSizeRule.EndValue** element.

```
<xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
```

## 2.232.8 MapSizeRule.LegendName

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSizeRule.LegendName** element specifies the name of the legend instance in which the rule distribution is displayed. The **MapSizeRule.LegendName** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). If this element is not present, the rule distribution MUST NOT be displayed in any legend.

Following is the parent element of the **MapSizeRule.LegendName** element.

Parent elements
<a href="#">MapSizeRule</a>

The following is the XML Schema definition of the **MapSizeRule.LegendName** element.

```
<xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
```

## 2.232.9 MapSizeRule.LegendText

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSizeRule.LegendText** element specifies the label to display for the map distribution in the legend instance that is specified by [MapSizeRule.LegendName](#). The **MapSizeRule.LegendText** element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as the start value of the bucket, followed by a hyphen (-), followed by the end value of the bucket.

Following is the parent element of the **MapSizeRule.LegendText** element.

Parent elements
<a href="#">MapSizeRule</a>

The following is the XML Schema definition of the **MapSizeRule.LegendText** element.

```
<xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
```

## 2.232.10 MapSizeRule.MapBuckets

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSizeRule.MapBuckets** element specifies a custom set of [MapBucket](#) instances to be used for data distribution. This element is optional.

This element MUST be specified if the value of the [MapSizeRule.DistributionType](#) element is "Custom". Otherwise, the value of the **MapSizeRule.MapBuckets** element is ignored. This element is of type [MapBuckets](#).

Following is the parent element of the **MapSizeRule.MapBuckets** element.

<b>Parent elements</b>
------------------------

<a href="#">MapSizeRule</a>
-----------------------------

The following is the XML Schema definition of the **MapSizeRule.MapBuckets** element.

```
<xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
```

### 2.232.11 MapSizeRule.StartSize

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapSizeRule.StartSize** element specifies the start size for the size range to be applied to **map points** or map lines based on the value of the [MapSizeRule.DataValue](#) element.

Following is the parent element of the **MapSizeRule.StartSize** element.

<b>Parent elements</b>
------------------------

<a href="#">MapSizeRule</a>
-----------------------------

The following is the XML Schema definition of the **MapSizeRule.StartSize** element.

```
<xsd:element name="StartSize" type="xsd:string" minOccurs="1" />
```

### 2.232.12 MapSizeRule.StartValue

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapSizeRule.StartValue** element specifies the start value for a [MapSizeRule](#) distribution. The **MapSizeRule.StartValue** element is optional.

This element is ignored if the value of the [MapColorPaletteRule.DistributionType](#) element is "Custom" or if the value of the [MapColorPaletteRule.DataValue](#) element is not scalar. Otherwise, if the **MapSizeRule.StartValue** element is not specified, its value MUST be automatically calculated.

Following is the parent element of the **MapSizeRule.StartValue** element.

<b>Parent elements</b>
------------------------

<a href="#">MapSizeRule</a>
-----------------------------

The following is the XML Schema definition of the **MapSizeRule.StartValue** element.

```
<xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
```

## 2.233 MapLines

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLines** element specifies an ordered list of [MapLine](#) elements in a map **line layer**. The **MapLines** element MUST contain at least one **MapLine** element.

The following are the parent and child elements of the **MapLines** element.

Parent elements
<a href="#">MapLineLayer</a>

Child elements
<a href="#">MapLines.MapLine</a>

The following is the XML Schema definition of the **MapLines** element.

```
<xsd:complexType name="MapLinesType">
  <xsd:sequence>
    <xsd:element name="MapLine" type="MapLineType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.233.1 MapLines.MapLine

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLines.MapLine** element specifies a **line**-based **spatial element** (for example, a street, a river, or a path) to embed in a [MapLineLayer](#). The **MapLines.MapLine** element MUST be specified at least once. This element is of type [MapLine](#).

Following is the parent element of the **MapLines.MapLine** element.

Parent elements
<a href="#">MapLines</a>

The following is the XML Schema definition of the **MapLines.MapLine** element.

```
<xsd:element name="MapLine" type="MapLineType" minOccurs="1"
  maxOccurs="unbounded" />
```

### 2.234 MapLine

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLine** element specifies a **line**-based **spatial element** (for example, a street, a river, or a path) to embed in a [MapLineLayer](#).

The following are the parent and child elements of the **MapLine** element.



<b>Parent elements</b>
<a href="#">MapLines</a>

<b>Child elements</b>
<a href="#">MapLine.MapFields</a>
<a href="#">MapLine.MapLineTemplate</a>
<a href="#">MapLine.UseCustomLineTemplate</a>
<a href="#">MapLine.VectorData</a>

The following is the XML Schema definition of the **MapLine** element.

```
<xsd:complexType name="MapLineType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSpatialElement Start-->
    <xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
    <xsd:element name="MapFields" type="MapFieldsType" minOccurs="0" />
    <!--MapSpatialElement End-->
    <xsd:element name="UseCustomLineTemplate" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapLineTemplate" type="MapLineTemplateType"
      minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.234.1 MapLine.MapFields

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLine.MapFields** element specifies a collection of [MapField](#) elements to associate with a map line. The **MapLine.MapFields** element is optional. This element is of type [MapFields](#).

Following is the parent element of the **MapLine.MapFields** element.

<b>Parent elements</b>
<a href="#">MapLine</a>

The following is the XML Schema definition of the **MapLine.MapFields** element.

```
<xsd:element name="MapFields" type="MapFieldsType" minOccurs="0" />
```

### 2.234.2 MapLine.MapLineTemplate

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLine.MapLineTemplate** element specifies a custom template to be applied to a map line. This element is optional.

This element is ignored if the value of the [MapLine.UseCustomLineTemplate](#) element is false. The **MapLine.MapLineTemplate** element MUST be specified if the value of the **MapLine.UseCustomLineTemplate** element is true. This element is of type [MapLineTemplate](#).

Following is the parent element of the **MapLine.MapLineTemplate** element.

Parent elements
<a href="#">MapLine</a>

The following is the XML Schema definition of the **MapLine.MapLineTemplate** element.

```
<xsd:element name="MapLineTemplate" type="MapLineTemplateType" minOccurs="0" />
```

### 2.234.3 MapLine.UseCustomLineTemplate

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLine.UseCustomLineTemplate** element specifies whether a [MapLine.MapLineTemplate](#) element will be applied to the parent map line. The **MapLine.UseCustomLineTemplate** element is optional.

If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapLine.UseCustomLineTemplate** element.

Parent elements
<a href="#">MapLine</a>

The following is the XML Schema definition of the **MapLine.UseCustomLineTemplate** element.

```
<xsd:element name="UseCustomLineTemplate" type="xsd:string" minOccurs="0" />
```

### 2.234.4 MapLine.VectorData

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLine.VectorData** element specifies the vector data for a map line. The **MapLine.VectorData** element MUST be specified. The value of this element MUST be a base64-encoded string of **Well-Known Binary** [\[ISO19125-2\]](#) format.

Following is the parent element of the **MapLine.VectorData** element.

Parent elements
<a href="#">MapLine</a>

The following is the XML Schema definition of the **MapLine.VectorData** element.

```
<xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
```

## 2.235 MapFields

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapFields** element specifies a collection of [MapField](#) elements. The **MapFields** element MUST contain at least one **MapField** element.

The following are the parent elements and child element of the **MapFields** element.

Parent elements
<a href="#">MapLine</a>
<a href="#">MapPoint</a>
<a href="#">MapPolygon</a>

Child elements
<a href="#">MapFields.MapField</a>

The following is the XML Schema definition of the **MapFields** element.

```
<xsd:complexType name="MapFieldsType">
  <xsd:sequence>
    <xsd:element name="MapField" type="MapFieldType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.235.1 MapFields.MapField

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapFields.MapField** element specifies a name and a value to be associated with a [MapPolygon](#), a [MapPoint](#), or a [MapLine](#). This element MUST be specified. This element is of type [MapField](#).

Following is the parent element of the **MapFields.MapField** element.

Parent elements
<a href="#">MapFields</a>

The following is the XML Schema definition of the **MapFields.MapField** element.

```
<xsd:element name="MapField" type="MapFieldType" minOccurs="1"
  maxOccurs="unbounded" />
```

## 2.236 MapField

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapField** element specifies a name and a value to be associated with a [MapPolygon](#), a [MapPoint](#), or a [MapLine](#).

The following are the parent and child elements of the **MapField** element.

Parent elements
<a href="#">MapFields</a>

Child elements
<a href="#">MapField.Name</a>
<a href="#">MapField.Value</a>

The following is the XML Schema definition of the **MapField** element.

```
<xsd:complexType name="MapFieldType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.236.1 MapField.Name

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapField.Name** element specifies the name of a [MapField](#). The **MapField.Name** element MUST be specified, and its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Following is the parent element of the **MapField.Name** element.

Parent elements
<a href="#">MapField</a>

The following is the XML Schema definition of the **MapField.Name** element.

```
<xsd:element name="Name" type="xsd:string" minOccurs="1" />
```

### 2.236.2 MapField.Value

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapField.Value** element specifies the value for a [MapField](#). The **MapField.Value** element MUST be specified, and its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Following is the parent element of the **MapField.Value** element.

Parent elements
MapField

The following is the XML Schema definition of the **MapField.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1" />
```

## 2.237 MapLineTemplate

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineTemplate** element specifies a template to be applied to map lines in a **map** line layer.

The following are the parent and child elements of the **MapLineTemplate** element.

Parent elements
<a href="#">MapLine</a>
<a href="#">MapLineLayer</a>

Child elements
<a href="#">MapLineTemplate.ActionInfo</a>
<a href="#">MapLineTemplate.DataElementLabel</a>
<a href="#">MapLineTemplate.DataElementName</a>
<a href="#">MapLineTemplate.DataElementOutput</a>
<a href="#">MapLineTemplate.Hidden</a>
<a href="#">MapLineTemplate.Label</a>
<a href="#">MapLineTemplate.LabelPlacement</a>
<a href="#">MapLineTemplate.OffsetX</a>
<a href="#">MapLineTemplate.OffsetY</a>
<a href="#">MapLineTemplate.Style</a>
<a href="#">MapLineTemplate.ToolTip</a>
<a href="#">MapLineTemplate.Width</a>

The following is the XML Schema definition of the **MapLineTemplate** element.

```
<xsd:complexType name="MapLineTemplateType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSpatialElementTemplateType Start-->
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetY" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
  </choice>
</complexType>
```

```

<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="DataElementLabel" type="xsd:string" minOccurs="0" />
<!--MapSpatialElementTemplateType End-->
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
<xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.237.1 MapLineTemplate.ActionInfo

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLineTemplate.ActionInfo** element specifies a collection of actions to be assigned to map lines in the parent layer. The **MapLineTemplate.ActionInfo** element is optional. This element is of type [ActionInfo](#).

Following is the parent element of the **MapLineTemplate.ActionInfo** element.

<b>Parent elements</b>
<a href="#">MapLineTemplate</a>

The following is the XML Schema definition of the **MapLineTemplate.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.237.2 MapLineTemplate.DataElementLabel

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLineTemplate.DataElementLabel** element specifies the label for the [MapLineTemplate](#) to use for a data element or attribute. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as the value of the [MapLineTemplate.Label](#) property.

Following is the parent element of the **MapLineTemplate.DataElementLabel** element.

<b>Parent elements</b>
MapLineTemplate

The following is the XML Schema definition of the **MapLineTemplate.DataElementLabel** element.

```
<xsd:element name="DataElementLabel" type="xsd:string" minOccurs="0" />
```

### 2.237.3 MapLineTemplate.DataElementName

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineLayer.DataElementName** element specifies the name for the [MapLineTemplate](#) to use for a data element or attribute. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as "MapDataRow".

Following is the parent element of the **MapLineLayer.DataElementName** element.

Parent elements
MapLineTemplate

The following is the XML Schema definition of the **MapLineLayer.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.237.4 MapLineTemplate.DataElementOutput

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineTemplate.DataElementOutput** element specifies whether a [MapLineTemplate](#) element will appear in a **data rendering**. The **MapLineTemplate.DataElementOutput** element is optional. If this element is present, its value MUST be one of the following:

**Output:** Specifies that the item appears in the data rendering output.

**NoOutput:** Specifies that the item does not appear in the data rendering output.

If this element is not present, its value is interpreted as "Output".

Following is the parent element of the **MapLineTemplate.DataElementOutput** element.

Parent elements
MapLineTemplate

The following is the XML Schema definition of the **MapLineTemplate.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.237.5 MapLineTemplate.Hidden

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineTemplate.Hidden** element specifies whether map lines to which this [MapLineTemplate](#) is applied are hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#)

([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapLineTemplate.Hidden** element.

Parent elements
<a href="#">MapLineTemplate</a>

The following is the XML Schema definition of the **MapLineTemplate.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

## 2.237.6 MapLineTemplate.Label

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineTemplate.Label** element specifies the label text for the map lines to which this [MapLineTemplate](#) is applied. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **MapLineTemplate.Label** element.

Parent elements
<a href="#">MapLineTemplate</a>

The following is the XML Schema definition of the **MapLineTemplate.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

## 2.237.7 MapLineTemplate.LabelPlacement

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineTemplate.LabelPlacement** element specifies the label placement for a [MapLineLayer](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Above:** Specifies that the label is above the [MapLine](#).

**Center:** Specifies that the label is centered on the map line.

**Below:** Specifies that the label is below the map line.

If this element is not present, its value is interpreted as "Above".

Following is the parent element of the **MapLineTemplate.LabelPlacement** element.

Parent elements
<a href="#">MapLineTemplate</a>



The following is the XML Schema definition of the **MapLineTemplate.LabelPlacement** element.

```
<xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
```

## 2.237.8 MapLineTemplate.OffsetX

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineTemplate.OffsetX** element specifies the X offset in **map coordinates** for map lines to which this [MapLineTemplate](#) is applied. This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapLineTemplate.OffsetX** element.

Parent elements
MapLineTemplate

The following is the XML Schema definition of the **MapLineTemplate.OffsetX** element.

```
<xsd:element name="OffsetX" type="xsd:string" minOccurs="0" />
```

## 2.237.9 MapLineTemplate.OffsetY

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineTemplate.OffsetY** element specifies the Y offset in **map coordinates** for map lines to which this [MapLineTemplate](#) is applied. This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapLineTemplate.OffsetY** element.

Parent elements
MapLineTemplate

The following is the XML Schema definition of the **MapLineTemplate.OffsetY** element.

```
<xsd:element name="OffsetY" type="xsd:string" minOccurs="0" />
```

## 2.237.10 MapLineTemplate.Style

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLineTemplate.Style** element specifies style information for map lines to which this [MapLineTemplate](#) is applied. This element is optional. This element is of type [Style](#).

Following is the parent element of the **MapLineTemplate.Style** element.

Parent elements
MapLineTemplate

The following is the XML Schema definition of the **MapLineTemplate.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.237.11 MapLineTemplate.ToolTip

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLineTemplate.ToolTip** element specifies the tooltip text for map lines to which this [MapLineTemplate](#) is applied. The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **MapLineTemplate.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **MapLineTemplate.ToolTip** element.

Parent elements
MapLineTemplate

The following is the XML Schema definition of the **MapLineTemplate.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

### 2.237.12 MapLineTemplate.Width

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLineTemplate.Width** element specifies the width of the map lines to which this [MapLineTemplate](#) is applied. This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 3.75pt.

Following is the parent element of the **MapLineTemplate.Width** element.

Parent elements
MapLineTemplate

The following is the XML Schema definition of the **MapLineTemplate.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
```

## 2.238 MapShapefile

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapShapefile** element specifies a file-based source for spatial data and optional non-spatial data.

The following are the parent elements and child elements of the **MapShapefile** element.

Parent elements
<a href="#">MapLineLayer</a>
<a href="#">MapPointLayer</a>
<a href="#">MapPolygonLayer</a>

Child elements
<a href="#">MapShapefile.MapFieldNames</a>
<a href="#">MapShapefile.Source</a>

The following is the XML Schema definition of the **MapShapefile** element.

```
<xsd:complexType name="MapShapefileType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="MapFieldNames" type="MapFieldNamesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.238.1 MapShapefile.MapFieldNames

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapShapefile.MapFieldNames** element specifies a collection of non-spatial data fields to be imported from the **shapefile**. This element is optional. This element is of type [MapFieldNames](#).

Following is the parent element of the **MapShapefile.MapFieldNames** element.

Parent elements
<a href="#">MapShapefile</a>

The following is the XML Schema definition of the **MapShapefile.MapFieldNames** element.

```
<xsd:element name="MapFieldNames" type="MapFieldNamesType" minOccurs="0" />
```

### 2.238.2 MapShapefile.Source

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapShapefile.Source** element specifies the location of a **shapefile** to be imported. This element **MUST** be specified.

The location for the shapefile can be an **absolute path**, a **relative path**, or a URL. Relative paths start in the same folder as the report. The value of this element **MUST** be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **MapShapefile.Source** element.

<b>Parent elements</b>
<a href="#">MapShapefile</a>

The following is the XML Schema definition of the **MapShapefile.Source** element.

```
<xsd:element name="Source" type="xsd:string" minOccurs="1" />
```

## 2.239 MapFieldNames

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapFieldNames** element specifies a collection of field names to be imported from a **shapefile** or from a SQL Server dataset. This element **MUST** contain at least one [MapFieldNames.MapFieldName](#) element.

The following are the parent elements and child element of the **MapFieldNames** element.

<b>Parent elements</b>
<a href="#">MapShapefile</a>
<a href="#">MapSpatialDataSet</a>

<b>Child elements</b>
MapFieldNames.MapFieldName

The following is the XML Schema definition of the **MapFieldNames** element.

```
<xsd:complexType name="MapFieldNamesType">
  <xsd:sequence>
    <xsd:element name="MapFieldName" type="xsd:string" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.239.1 MapFieldNames.MapFieldName

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapFieldNames.MapFieldName** element specifies a field name to be imported from a **shapefile** or from a SQL Server dataset. This element **MUST** be specified. The value of this element **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **MapFieldNames.MapFieldName** element.

<b>Parent elements</b>
<a href="#">MapFieldNames</a>

The following is the XML Schema definition of the **MapFieldNames.MapFieldName** element.

```
<xsd:element name="MapFieldName" type="xsd:string" minOccurs="1"
maxOccurs="unbounded" />
```

## 2.240 MapSpatialDataRegion

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSpatialDataRegion** element specifies a data region-based source for spatial data.

The following are the parent elements and child element of the **MapSpatialDataRegion** element.

Parent elements
<a href="#">MapLineLayer</a>
<a href="#">MapPointLayer</a>
<a href="#">MapPolygonLayer</a>

Child elements
<a href="#">MapSpatialDataRegion.VectorData</a>

The following is the XML Schema definition of the **MapSpatialDataRegion** element.

```
<xsd:complexType name="MapSpatialDataRegionType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.240.1 MapSpatialDataRegion.VectorData

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSpatialDataRegion.VectorData** element specifies how vector data for spatial elements is imported from the data region. This element **MUST** be specified as an expression that evaluates to either **SqlGeometry** or **SqlGeography**.

Following is the parent element of the **MapSpatialDataRegion.VectorData** element.

Parent elements
<a href="#">MapSpatialDataRegion</a>

The following is the XML Schema definition of the **MapSpatialDataRegion.VectorData** element.

```
<xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
```

## 2.241 MapSpatialDataSet

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSpatialDataSet** element specifies a dataset-based source for spatial and optional non-spatial data.

The following are the parent elements and child elements of the **MapSpatialDataSet** element.

Parent elements
<a href="#">MapLineLayer</a>
<a href="#">MapPointLayer</a>
<a href="#">MapPolygonLayer</a>

Child elements
<a href="#">MapSpatialDataSet.DataSetName</a>
<a href="#">MapSpatialDataSet.MapFieldNames</a>
<a href="#">MapSpatialDataSet.SpatialField</a>

The following is the XML Schema definition of the **MapSpatialDataSet** element.

```
<xsd:complexType name="MapSpatialDataSetType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="1" />
    <xsd:element name="SpatialField" type="xsd:string" minOccurs="1" />
    <xsd:element name="MapFieldNames" type="MapFieldNamesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.241.1 MapSpatialDataSet.DataSetName

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSpatialDataSet.DataSetName** element specifies the name of the dataset from which spatial data is extracted. This element **MUST** be specified. If filters were specified for this dataset, the filters is ignored.

Following is the parent element of the **MapSpatialDataSet.DataSetName** element.

Parent elements
<a href="#">MapSpatialDataSet</a>

The following is the XML Schema definition of the **MapSpatialDataSet.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="1" />
```

### 2.241.2 MapSpatialDataSet.MapFieldNames

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSpatialDataSet.MapFieldNames** element specifies a collection of names for the fields to be imported from the dataset. This element is optional. This element is of type [MapFieldNames](#).

Following is the parent element of the **MapSpatialDataSet.MapFieldNames** element.

Parent elements
<a href="#">MapSpatialDataSet</a>

The following is the XML Schema definition of the **MapSpatialDataSet.MapFieldNames** element.

```
<xsd:element name="MapFieldNames" type="MapFieldNamesType" minOccurs="0" />
```

### 2.241.3 MapSpatialDataSet.SpatialField

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapSpatialDataSet.SpatialField** element specifies the name for the spatial field that is to be imported from the data source. This element **MUST** be specified, and it **MUST** be of type **SqlGeometry** or **SqlGeography** [\[OGCSPEC\]](#).

Following is the parent element of the **MapSpatialDataSet.SpatialField** element.

Parent elements
<a href="#">MapSpatialDataSet</a>

The following is the XML Schema definition of the **MapSpatialDataSet.SpatialField** element.

```
<xsd:element name="SpatialField" type="xsd:string" minOccurs="1" />
```

### 2.242 MapPointLayer

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointLayer** element specifies a point-based layer to be drawn in a Map.

The following are the parent element, attribute, and child elements of the **MapPointLayer** element.

Parent elements
<a href="#">MapLayers</a>

Attributes
<a href="#">MapPointLayer.Name</a>

Child elements
<a href="#">MapPointLayer.DataElementName</a>

Child elements
<a href="#">MapPointLayer.DataElementOutput</a>
<a href="#">MapPointLayer.MapBindingFieldPairs</a>
<a href="#">MapPointLayer.MapDataRegionName</a>
<a href="#">MapPointLayer.MapFieldDefinitions</a>
<a href="#">MapPointLayer.MapPointRules</a>
<a href="#">MapPointLayer.MapPoints</a>
<a href="#">MapPointLayer.MapMarkerTemplate</a>
<a href="#">MapPointLayer.MapShapefile</a>
<a href="#">MapPointLayer.MapSpatialDataSet</a>
<a href="#">MapPointLayer.MapSpatialDataRegion</a>
<a href="#">MapPointLayer.MaximumZoom</a>
<a href="#">MapPointLayer.MinimumZoom</a>
<a href="#">MapPointLayer.Transparency</a>
<a href="#">MapPointLayer.VisibilityMode</a>

The following is the XML Schema definition of the **MapPointLayer** element.

```

<xsd:complexType name="MapPointLayerType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapLayerStart-->
    <xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <!--MapLayerEnd-->
    <!--MapVectorLayerStart-->
    <xsd:element name="MapDataRegionName" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType"
      minOccurs="0" />
    <xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType"
      minOccurs="0" />
    <xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
    <xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType"
      minOccurs="0" />
    <xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType"
      minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapVectorLayerEnd-->
    <xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType"
      minOccurs="0" />
    <xsd:element name="MapPointRules" type="MapPointRulesType" minOccurs="0" />
    <xsd:element name="MapPoints" type="MapPointsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
</complexType>

```



```

</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.242.1 MapPointLayer.Name

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPointLayer.Name** attribute specifies a name for a [MapPointLayer](#) element. The **MapPointLayer.Name** attribute MUST be specified. The value of this attribute MUST be a case-sensitive [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier**.

Following is the parent element of the **MapPointLayer.Name** attribute.

Parent elements
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.242.2 MapPointLayer.DataElementName

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPointLayer.DataElementName** element specifies the name for a [MapPointLayer](#) to use for the data element or attribute. This element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as the name attribute of the map line layer.

Following is the parent element of the **MapPointLayer.DataElementName** element.

Parent elements
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.242.3 MapPointLayer.DataElementOutput

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPointLayer.DataElementOutput** element specifies whether this [MapPointLayer](#) element will appear in a **data rendering**. The **MapPointLayer.DataElementOutput** element is optional.

If this element is present, its value MUST be one of the following:

**Output:** Specifies that the item appears in the data rendering output.

**NoOutput:** Specifies that the item does not appear in the data rendering output.

If this element is not present, its value is interpreted as "Output".

Following is the parent element of the **MapPointLayer.DataElementOutput** element.

Parent elements
<a href="#">MapPointLayer</a>

The following is the XML Schema definition of the **MapPointLayer.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

#### 2.242.4 MapPointLayer.MapBindingFieldPairs

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPointLayer.MapBindingFieldPairs** element specifies a collection of [MapBindingFieldPair](#) elements to be used to bind the map items in the parent map **point layer** to the data region that is associated with the layer. The **MapPointLayer.MapBindingFieldPairs** element is optional.

This element MUST be specified if the value of the **MapPointLayer.MapDataRegionName** element is specified and the value of the **MapPointLayer.MapSpatialDataRegion** element is not specified. Otherwise, the value of the **MapPointLayer.MapBindingFieldPairs** element is ignored. This element is of type [MapBindingFieldPairs](#).

Following is the parent element of the **MapPointLayer.MapBindingFieldPairs** element.

Parent elements
<a href="#">MapPointLayer</a>

The following is the XML Schema definition of the **MapPointLayer.MapBindingFieldPairs** element.

```
<xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType" minOccurs="0" />
```

#### 2.242.5 MapPointLayer.MapDataRegionName

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPointLayer.MapDataRegionName** element specifies the data region from which to consume data. This element is optional. This element MUST be specified if the value of the [MapPointLayer.MapSpatialDataRegion](#) element is specified.

Following is the parent element of the **MapPointLayer.MapDataRegionName** element.

Parent elements
<a href="#">MapPointLayer</a>

The following is the XML Schema definition of the **MapPointLayer.MapDataRegionName** element.

```
<xsd:element name="MapDataRegionName" type="xsd:string" minOccurs="0" />
```

## 2.242.6 MapPointLayer.MapFieldDefinitions

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointLayer.MapFieldDefinitions** element specifies a collection of [MapFieldDefinition](#) elements in the parent **point layer**. This collection describes the metadata for the fields that are associated with the **map points**. The **MapPointLayer.MapFieldDefinitions** element is optional.

This element is ignored if the value of the **MapPointLayer.MapPoints** element is not specified. The **MapPointLayer.MapFieldDefinitions** element is of type [MapFieldDefinitions](#).

Following is the parent element of the **MapPointLayer.MapFieldDefinitions** element.

Parent elements
<a href="#">MapPointLayer</a>

The following is the XML Schema definition of the **MapPointLayer.MapFieldDefinitions** element.

```
<xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType" minOccurs="0" />
```

## 2.242.7 MapPointLayer.MapPointRules

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointLayer.MapPointRules** element specifies a set of rules to be applied to the **map points** in the parent layer. This element is optional. This element is of type [MapPointRules](#).

Following is the parent element of the **MapPointLayer.MapPointRules** element.

Parent elements
<a href="#">MapPointLayer</a>

The following is the XML Schema definition of the **MapPointLayer.MapPointRules** element.

```
<xsd:element name="MapPointRules" type="MapPointRulesType" minOccurs="0" />
```

## 2.242.8 MapPointLayer.MapPoints

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointLayer.MapPoints** element specifies a collection of embedded **map points**. This element is optional. This element is of type [MapPoints](#).

Following is the parent element of the **MapPointLayer.MapPoints** element.

<b>Parent elements</b>
------------------------

<a href="#">MapPointLayer</a>
-------------------------------

The following is the XML Schema definition of the **MapPointLayer.MapPoints** element.

```
<xsd:element name="MapPoints" type="MapPointsType" minOccurs="0" />
```

## 2.242.9 MapPointLayer.MapMarkerTemplate

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointLayer.MapMarkerTemplate** element specifies a **marker template** to be applied for points in the parent **map layer**. This element is optional. This element is of type [MapMarkerTemplate](#).

Following is the parent element of the **MapPointLayer.MapMarkerTemplate** element.

<b>Parent elements</b>
------------------------

<a href="#">MapPointLayer</a>
-------------------------------

The following is the XML Schema definition of the **MapPointLayer.MapMarkerTemplate** element.

```
<xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType" minOccurs="0" />
```

## 2.242.10 MapPointLayer.MapSpatialDataRegion

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointLayer.MapSpatialDataRegion** element specifies a map data region as the source for the **map points** for the parent layer. This element is optional. This element is ignored if the value of the [MapPointLayer.MapPoints](#) element is specified.

The **MapPointLayer.MapSpatialDataRegion** element cannot be present if either of the following elements is present:

- [MapPointLayer.MapSpatialDataSet](#)
- [MapPointLayer.MapShapefile](#)

The **MapPointLayer.MapSpatialDataRegion** element is of type [MapSpatialDataRegion](#).

Following is the parent element of the **MapPointLayer.MapSpatialDataRegion** element.

<b>Parent elements</b>
------------------------

<a href="#">MapPointLayer</a>
-------------------------------

The following is the XML Schema definition of the **MapPointLayer.MapSpatialDataRegion** element.

```
<xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType" minOccurs="0" />
```

## 2.242.11 MapPointLayer.MapShapefile

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointLayer.MapShapefile** element specifies a **shapefile** as the source for the map points for the parent layer. This element is optional. This element is ignored if the value of the [MapPointLayer.MapPoints](#) element is specified.

The **MapPointLayer.MapShapefile** element cannot be present if either of the following elements is present:

- [MapPointLayer.MapSpatialDataSet](#)
- [MapPointLayer.MapSpatialDataRegion](#)

The **MapPointLayer.MapShapefile** element is of type [MapShapefile](#).

Following is the parent element of the **MapPointLayer.MapShapefile** element.

Parent elements
<a href="#">MapPointLayer</a>

The following is the XML Schema definition of the **MapPointLayer.MapShapefile** element.

```
<xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
```

## 2.242.12 MapPointLayer.MapSpatialDataSet

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointLayer.MapSpatialDataSet** element specifies a SQL Server dataset as the source for the map points for the parent layer. This element is optional. This element is ignored if the value of the [MapPointLayer.MapPoints](#) element is specified.

The **MapPointLayer.MapSpatialDataSet** element cannot be present if either of the following elements is present:

- [MapPointLayer.MapSpatialDataRegion](#)
- [MapPointLayer.MapShapefile](#)

The **MapPointLayer.MapSpatialDataSet** element is of type [MapSpatialDataSet](#).

Following is the parent element of the **MapPointLayer.MapSpatialDataSet** element.

Parent elements
<a href="#">MapPointLayer</a>

The following is the XML Schema definition of the **MapPointLayer.MapSpatialDataSet** element.

```
<xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType" minOccurs="0" />
```

## 2.242.13 MapPointLayer.MaximumZoom

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointLayer.MaximumZoom** element specifies the maximum zoom level at which a parent layer is visible. This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 200. This element is ignored if the value of the [MapPointLayer.VisibilityMode](#) element is not "ZoomBased".

Following is the parent element of the **MapPointLayer.MaximumZoom** element.

Parent elements
<a href="#">MapPointLayer</a>

The following is the XML Schema definition of the **MapPointLayer.MaximumZoom** element.

```
<xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
```

## 2.242.14 MapPointLayer.MinimumZoom

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointLayer.MinimumZoom** element specifies the minimum zoom level at which a parent layer is visible. This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 50. This element is ignored if the value of the [MapPointLayer.VisibilityMode](#) element is not "ZoomBased".

Following is the parent element of the **MapPointLayer.MinimumZoom** element.

Parent elements
<a href="#">MapPointLayer</a>

The following is the XML Schema definition of the **MapPointLayer.MinimumZoom** element.

```
<xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
```

## 2.242.15 MapPointLayer.Transparency

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointLayer.Transparency** element specifies the transparency for the map points in the parent layer as a percentage. This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 100.

Following is the parent element of the **MapPointLayer.Transparency** element.

Parent elements
<a href="#">MapPointLayer</a>

The following is the XML Schema definition of the **MapPointLayer.Transparency** element.

```
<xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
```

## 2.242.16 MapPointLayer.VisibilityMode

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointLayer.VisibilityMode** element specifies the visibility mode for the map **point layer**. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Visible:** The layer is always visible.

**Hidden:** The layer is always hidden.

**ZoomBased:** The layer is shown in a zoom level range that is defined by **MapPointLayer.MaximumZoom** and **MapPointLayer.MinimumZoom**.

If the **MapPointLayer.VisibilityMode** element is not present, its value is interpreted as "Visible".

Following is the parent element of the **MapPointLayer.VisibilityMode** element.

Parent elements
<a href="#">MapPointLayer</a>

The following is the XML Schema definition of the **MapPointLayer.VisibilityMode** element.

```
<xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
```

## 2.243 MapMarkerTemplate

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerTemplate** element specifies a template to be applied to **map points** in a [MapPointLayer](#) or a [MapPolygonLayer](#).

The following are the parent and child elements of the **MapMarkerTemplate** element.

Parent elements
MapPointLayer
MapPolygonLayer

Child elements
<a href="#">MapMarkerTemplate.ActionInfo</a>
<a href="#">MapMarkerTemplate.DataElementLabel</a>
<a href="#">MapMarkerTemplate.DataElementName</a>

Child elements
<a href="#">MapMarkerTemplate.DataElementOutput</a>
<a href="#">MapMarkerTemplate.Hidden</a>
<a href="#">MapMarkerTemplate.Label</a>
<a href="#">MapMarkerTemplate.LabelPlacement</a>
<a href="#">MapMarkerTemplate.MapMarker</a>
<a href="#">MapMarkerTemplate.OffsetX</a>
<a href="#">MapMarkerTemplate.OffsetY</a>
<a href="#">MapMarkerTemplate.Size</a>
<a href="#">MapMarkerTemplate.Style</a>
<a href="#">MapMarkerTemplate.ToolTip</a>

The following is the XML Schema definition of the **MapMarkerTemplate** element.

```
<xsd:complexType name="MapMarkerTemplateType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSpatialElementTemplateType Start-->
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetY" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementLabel" type="xsd:string" minOccurs="0" />
    <!--MapSpatialElementTemplateType End-->
    <!--MapPointTemplateType Start-->
    <xsd:element name="Size" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
    <!--MapPointTemplateTypeType End-->
    <xsd:element name="MapMarker" type="MapMarkerType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.243.1 MapMarkerTemplate.ActionInfo

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapMarkerTemplate.ActionInfo** element specifies a collection of actions to be assigned to map points in the parent layer. This element is optional. This element is of type [ActionInfo](#).

Following is the parent element of the **MapMarkerTemplate.ActionInfo** element.



<b>Parent elements</b>
------------------------

<a href="#">MapMarkerTemplate</a>
-----------------------------------

The following is the XML Schema definition of the **MapMarkerTemplate.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

## 2.243.2 MapMarkerTemplate.DataElementLabel

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapMarkerTemplate.DataElementLabel** element specifies the label for the **marker template** to use for a data element or attribute. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as the value of the [MapMarkerTemplate.Label](#) property.

Following is the parent element of the **MapMarkerTemplate.DataElementLabel** element.

<b>Parent elements</b>
------------------------

<a href="#">MapMarkerTemplate</a>
-----------------------------------

The following is the XML Schema definition of the **MapMarkerTemplate.DataElementLabel** element.

```
<xsd:element name="DataElementLabel" type="xsd:string" minOccurs="0" />
```

## 2.243.3 MapMarkerTemplate.DataElementName

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapMarkerTemplate.DataElementName** element specifies a name for the **marker template** to use for a data element or attribute. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as "MapDataRow".

Following is the parent element of the **MapMarkerTemplate.DataElementName** element.

<b>Parent elements</b>
------------------------

<a href="#">MapMarkerTemplate</a>
-----------------------------------

The following is the XML Schema definition of the **MapMarkerTemplate.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

## 2.243.4 MapMarkerTemplate.DataElementOutput

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapMarkerTemplate.DataElementOutput** element specifies whether the **marker template** will appear in a **data rendering**. This element is optional.

If this element is present, its value MUST be one of the following:

**Output:** Specifies that the item appears in the data rendering output.

**NoOutput:** Specifies that the item does not appear in the data rendering output.

If this element is not present, its value is interpreted as "Output".

Following is the parent element of the **MapMarkerTemplate.DataElementOutput** element.

Parent elements
<a href="#">MapMarkerTemplate</a>

The following is the XML Schema definition of the **MapMarkerTemplate.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.243.5 MapMarkerTemplate.Hidden

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapMarkerTemplate.Hidden** element specifies whether the map points to which this **marker template** is applied are hidden. This element is optional.

If this element is present, its value MUST be a **Boolean** ([XMLSCHEMA2/21](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapMarkerTemplate.Hidden** element.

Parent elements
<a href="#">MapMarkerTemplate</a>

The following is the XML Schema definition of the **MapMarkerTemplate.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

## 2.243.6 MapMarkerTemplate.Label

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapMarkerTemplate.Label** element specifies the label text for the map points to which this **marker template** is applied. This element is optional.

If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **MapMarkerTemplate.Label** element.

Parent elements
<a href="#">MapMarkerTemplate</a>

The following is the XML Schema definition of the **MapMarkerTemplate.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

## 2.243.7 MapMarkerTemplate.LabelPlacement

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerTemplate.LabelPlacement** element specifies the label placement for the [MapPointLayer](#). This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Bottom:** Specifies that the label is positioned at the bottom of the map **point layer**.

**Top:** Specifies that the label is positioned at the top of the map point layer.

**Left:** Specifies that the label is positioned to the left of the map point layer.

**Right:** Specifies that the label is positioned to the right of the map point layer.

**Center:** Specifies that the label is positioned at the center of the map point layer.

If this element is not present, its value is interpreted as "Bottom".

Following is the parent element of the **MapMarkerTemplate.LabelPlacement** element.

Parent elements
<a href="#">MapMarkerTemplate</a>

The following is the XML Schema definition of the **MapMarkerTemplate.LabelPlacement** element.

```
<xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
```

## 2.243.8 MapMarkerTemplate.MapMarker

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerTemplate.MapMarker** element specifies the marker to be associated with the map points in the parent **map layer**. This element is optional. This element is of type [MapMarker](#).

Following is the parent element of the **MapMarkerTemplate.MapMarker** element.

<b>Parent elements</b>
------------------------

<a href="#">MapMarkerTemplate</a>
-----------------------------------

The following is the XML Schema definition of the **MapMarkerTemplate.MapMarker** element.

```
<xsd:element name="MapMarker" type="MapMarkerType" minOccurs="0" />
```

### 2.243.9 MapMarkerTemplate.OffsetX

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapMarkerTemplate.OffsetX** element specifies the X offset in **map coordinates** for **map points** to which this marker template is applied. This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapMarkerTemplate.OffsetX** element.

<b>Parent elements</b>
------------------------

<a href="#">MapMarkerTemplate</a>
-----------------------------------

The following is the XML Schema definition of the **MapMarkerTemplate.OffsetX** element.

```
<xsd:element name="OffsetX" type="xsd:string" minOccurs="0" />
```

### 2.243.10 MapMarkerTemplate.OffsetY

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapMarkerTemplate.OffsetY** element specifies the Y offset in **map coordinates** for **map points** to which this marker template is applied. This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapMarkerTemplate.OffsetY** element.

<b>Parent elements</b>
------------------------

<a href="#">MapMarkerTemplate</a>
-----------------------------------

The following is the XML Schema definition of the **MapMarkerTemplate.OffsetY** element.

```
<xsd:element name="OffsetY" type="xsd:string" minOccurs="0" />
```

### 2.243.11 MapMarkerTemplate.Size

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapMarkerTemplate.Size** element specifies the size of the map points to which this marker template is applied. This element is optional.

If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**. If this element is not present, its value is interpreted as 5.25pt.

Following is the parent element of the **MapMarkerTemplate.Size** element.

Parent elements
<a href="#">MapMarkerTemplate</a>

The following is the XML Schema definition of the **MapMarkerTemplate.Size** element.

```
<xsd:element name="Size" type="xsd:string" minOccurs="0" />
```

### 2.243.12 MapMarkerTemplate.Style

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerTemplate.Style** element specifies style information for the map points to which this marker template is applied. This element is optional and is of type [Style](#).

Following is the parent element of the **MapMarkerTemplate.Style** element.

Parent elements
<a href="#">MapMarkerTemplate</a>

The following is the XML Schema definition of the **MapMarkerTemplate.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.243.13 MapMarkerTemplate.ToolTip

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerTemplate.ToolTip** element specifies the tooltip text for the map points to which this marker template is applied. The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **MapMarkerTemplate.ToolTip** element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **MapMarkerTemplate.ToolTip** element.

Parent elements
<a href="#">MapMarkerTemplate</a>

The following is the XML Schema definition of the **MapMarkerTemplate.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

## 2.244 MapPointRules

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointRules** element specifies a set of rules to be applied to map points in the parent layer.

The following are the parent and child elements of the **MapPointRules** element.

Parent elements
<a href="#">MapPointLayer</a>
<a href="#">MapPolygonLayer</a>

Child elements
<a href="#">MapLineRules.MapColorPaletteRule</a>
<a href="#">MapLineRules.MapColorRangeRule</a>
<a href="#">MapLineRules.MapCustomColorRule</a>
<a href="#">MapPointRules.MapMarkerRule</a>
<a href="#">MapLineRules.MapSizeRule</a>

The following is the XML Schema definition of the **MapPointRules** element.

```
<xsd:complexType name="MapPointRulesType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType"
      minOccurs="0" />
    <xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType"
      minOccurs="0" />
    <xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType"
      minOccurs="0" />
    <xsd:element name="MapSizeRule" type="MapSizeRuleType" minOccurs="0" />
    <xsd:element name="MapMarkerRule" type="MapMarkerRuleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.244.1 MapPointRules.MapColorPaletteRule

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointRules.MapColorPaletteRule** element specifies a palette of colors for map points in the parent layer based on the value of the [MapColorPaletteRule.DataValue](#) element. The **MapPointRules.MapColorPaletteRule** element is optional. If this element is present, it overrides the [Style.Color](#) child element of the [MapMarkerTemplate](#) element.

The **MapPointRules.MapColorPaletteRule** element cannot be present if either of the following elements is present:

- [MapPointRules.MapColorRangeRule](#)
- [MapPointRules.MapCustomColorRule](#)

The **MapPointRules.MapColorPaletteRule** element is of type [MapColorPaletteRule](#).

Following is the parent element of the **MapPointRules.MapColorPaletteRule** element.

Parent elements
<a href="#">MapPointRules</a>

The following is the XML Schema definition of the **MapPointRules.MapColorPaletteRule** element.

```
<xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType" minOccurs="0" />
```

## 2.244.2 MapPointRules.MapColorRangeRule

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointRules.MapColorRangeRule** element specifies a color range to be used for coloring map points in the parent layer based on the value of the [MapColorRangeRule.DataValue](#) element. The **MapPointRules.MapColorRangeRule** element is optional. If this element is present, it overrides the [Style.Color](#) child element of the [MapMarkerTemplate](#) element.

The **MapPointRules.MapColorRangeRule** element cannot be present if either of the following elements is present:

- [MapPointRules.MapColorPaletteRule](#)
- [MapPointRules.MapCustomColorRule](#)

The **MapPointRules.MapColorRangeRule** element is of type [MapColorRangeRule](#).

Following is the parent element of the **MapPointRules.MapColorRangeRule** element.

Parent elements
<a href="#">MapPointRules</a>

The following is the XML Schema definition of the **MapPointRules.MapColorRangeRule** element.

```
<xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType" minOccurs="0" />
```

## 2.244.3 MapPointRules.MapCustomColorRule

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointRules.MapCustomColorRule** element specifies a user-defined set of colors for map points in the parent layer based on the value of the [MapCustomColorRule.DataValue](#) element. The **MapPointRules.MapCustomColorRule** element is optional. If this element is present, it overrides the [Style.Color](#) child element of the [MapMarkerTemplate](#) element.

The **MapPointRules.MapCustomColorRule** element cannot be present if either of the following elements is present:

- [MapPointRules.MapColorPaletteRule](#)
- [MapPointRules.MapColorRangeRule](#)

The **MapPointRules.MapCustomColorRule** element is of type [MapCustomColorRule](#).

Following is the parent element of the **MapPointRules.MapCustomColorRule** element.

Parent elements
<a href="#">MapPointRules</a>

The following is the XML Schema definition of the **MapPointRules.MapCustomColorRule** element.

```
<xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType" minOccurs="0" />
```

#### 2.244.4 MapPointRules.MapMarkerRule

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointRules.MapMarkerRule** element specifies a set of markers to be applied to map points in the parent layer based on the value of the [MapMarkerRule.DataValue](#) element. The **MapPointRules.MapMarkerRule** element is optional.

If this element is present, it overrides [MapMarkerTemplate.MapMarker](#). The **MapPointRules.MapMarkerRule** element is of type [MapMarkerRule](#).

Following is the parent element of the **MapPointRules.MapMarkerRule** element.

Parent elements
<a href="#">MapPointRules</a>

The following is the XML Schema definition of the **MapPointRules.MapMarkerRule** element.

```
<xsd:element name="MapMarkerRule" type="MapMarkerRuleType" minOccurs="0" />
```

#### 2.244.5 MapPointRules.MapSizeRule

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPointRules.MapSizeRule** element specifies a size range to be used for sizing map points in the parent layer based on the value of the [MapSizeRule.DataValue](#) element. The **MapPointRules.MapSizeRule** element is optional.

If this element is present, it overrides [MapMarkerTemplate.Size](#). The **MapPointRules.MapSizeRule** element is of type [MapSizeRule](#).

Following is the parent element of the **MapPointRules.MapSizeRule** element.



<b>Parent elements</b>
<a href="#">MapPointRules</a>

The following is the XML Schema definition of the **MapPointRules.MapSizeRule** element.

```
<xsd:element name="MapSizeRule" type="MapSizeRuleType" minOccurs="0" />
```

## 2.245 MapMarkerRule

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapMarkerRule** element specifies a set of markers for map points in a [MapPolygonLayer](#) or in a [MapPointLayer](#) based on the value of the [MapMarkerRule.DataValue](#) element.

The following are the parent and child elements of the **MapMarkerRule** element.

<b>Parent elements</b>
<a href="#">MapPointRules</a>

Child elements
<a href="#">MapMarkerRule.BucketCount</a>
<a href="#">MapMarkerRule.DataElementName</a>
<a href="#">MapMarkerRule.DataElementOutput</a>
MapMarkerRule.DataValue
<a href="#">MapMarkerRule.DistributionType</a>
<a href="#">MapMarkerRule.EndValue</a>
<a href="#">MapMarkerRule.LegendName</a>
<a href="#">MapMarkerRule.LegendText</a>
<a href="#">MapMarkerRule.MapBuckets</a>
<a href="#">MapMarkerRule.MapMarkers</a>
<a href="#">MapMarkerRule.StartValue</a>

The following is the XML Schema definition of the **MapMarkerRule** element.

```
<xsd:complexType name="MapMarkerRuleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapAppearanceRule Start-->
    <xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
  </choice>
</complexType>
```

```

<xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<!--MapAppearanceRule End-->
<xsd:element name="MapMarkers" type="MapMarkersType" minOccurs="1" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

## 2.245.1 MapMarkerRule.BucketCount

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerRule.BucketCount** element specifies the number of buckets for the [MapMarkerRule](#) element. The **MapMarkerRule.BucketCount** element is optional.

If this element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2](#) section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 5. This element is ignored if the value of the [MapMarkerRule.DistributionType](#) element is "Custom" or if the value of the [MapMarkerRule.DataValue](#) element is not scalar.

Following is the parent element of the **MapMarkerRule.BucketCount** element.

Parent elements
<a href="#">MapMarkerRule</a>

The following is the XML Schema definition of the **MapMarkerRule.BucketCount** element.

```

<xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />

```

## 2.245.2 MapMarkerRule.DataElementName

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerRule.DataElementName** element specifies the name to use for the data element or attribute for the [MapMarkerRule.DataValue](#) element. The **MapMarkerRule.DataElementName** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#).

Following is the parent of the **MapMarkerRule.DataElementName** element.

Parent elements
<a href="#">MapMarkerRule</a>

The following is the XML Schema definition of the **MapMarkerRule.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.245.3 MapMarkerRule.DataElementOutput

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerRule.DataElementOutput** element indicates whether the parent [MapMarkerRule](#) element will appear in a **data rendering**. The **MapMarkerRule.DataElementOutput** element is optional.

If this element is present, its value MUST be one of the following:

**Output:** Specifies that the item appears in the data rendering output.

**NoOutput:** Specifies that the item does not appear in the data rendering output.

If this element is not present, its value is interpreted as "Output".

Following is the parent element of the **MapMarkerRule.DataElementOutput** element.

Parent elements
<a href="#">MapMarkerRule</a>

The following is the XML Schema definition of the **MapMarkerRule.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.245.4 MapMarkerRule.DataValue

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerRule.DataValue** element specifies an expression to be evaluated on the scope of the map item in a [MapPolygonLayer](#), a [MapPointLayer](#), or a [MapLineLayer](#). This element is optional.

The specified markers are assigned to the map items based on this element. If this element is not present, each map item is assigned a distinct marker type.

Following is the parent element of the **MapMarkerRule.DataValue** element.

Parent elements
<a href="#">MapMarkerRule</a>

The following is the XML Schema definition of the **MapMarkerRule.DataValue** element.

```
<xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
```

## 2.245.5 MapMarkerRule.DistributionType

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerRule.DistributionType** element specifies the distribution type for a [MapMarkerRule](#) instance. The **MapMarkerRule.DistributionType** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Optimal:** The optimal distribution is applied.

**EqualInterval:** The equal-interval distribution is applied.

**EqualDistribution:** The equal distribution is applied.

**Custom:** The user-defined distribution is applied.

If this element is not present, its value is interpreted as "Optimal".

Following is the parent element of the **MapMarkerRule.DistributionType** element.

Parent elements
MapMarkerRule

The following is the XML Schema definition of the **MapMarkerRule.DistributionType** element.

```
<xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
```

## 2.245.6 MapMarkerRule.EndValue

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerRule.EndValue** element specifies the end value for a [MapMarkerRule](#) distribution. The **MapMarkerRule.EndValue** element is optional.

This element is ignored if the value of the [MapMarkerRule.DistributionType](#) element is "Custom" or if the value of the [MapMarkerRule.DataValue](#) element is not scalar. Otherwise, if the **MapMarkerRule.EndValue** element is not present, its value MUST be automatically calculated.

Following is the parent element of the **MapMarkerRule.EndValue** element.

Parent elements
MapMarkerRule

The following is the XML Schema definition of the **MapMarkerRule.EndValue** element.

```
<xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
```

## 2.245.7 MapMarkerRule.LegendName

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerRule.LegendName** element specifies the name of the legend instance in which the rule distribution is displayed. The **MapMarkerRule.LegendName** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1). If this element is not present, the rule distribution MUST NOT be displayed in any legend.

Following is the parent element of the **MapMarkerRule.LegendName** element.

Parent elements
<a href="#">MapMarkerRule</a>

The following is the XML Schema definition of the **MapMarkerRule.LegendName** element.

```
<xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
```

## 2.245.8 MapMarkerRule.LegendText

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerRule.LegendText** element specifies the label to be displayed for the map distribution in the legend instance that is specified by [MapCustomColorRule.LegendName](#). The **MapMarkerRule.LegendText** element is optional.

If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as the start value of the bucket, followed by a hyphen (-), followed by the end value of the bucket.

Following is the parent element of the **MapMarkerRule.LegendText** element.

Parent elements
<a href="#">MapMarkerRule</a>

The following is the XML Schema definition of the **MapMarkerRule.LegendText** element.

```
<xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
```

## 2.245.9 MapMarkerRule.MapBuckets

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerRule.MapBuckets** element specifies a custom set of [MapBucket](#) instances to be used for a data distribution. The **MapMarkerRule.MapBuckets** element is optional.

This element MUST be specified if the value of the [MapMarkerRule.DistributionType](#) element is specified as "Custom". Otherwise, the value of the **MapMarkerRule.MapBuckets** element is ignored. This element is of type [MapBuckets](#).

Following is the parent element of the **MapMarkerRule.MapBuckets** element.

Parent elements
<a href="#">MapMarkerRule</a>

The following is the XML Schema definition of the **MapMarkerRule.MapBuckets** element.

```
<xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
```

## 2.245.10 MapMarkerRule.MapMarkers

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerRule.MapMarkers** element specifies a collection of markers to be applied to map points in the layer based on the value of the [MapMarkerRule.DataValue](#) element. The **MapMarkerRule.MapMarkers** element MUST be specified. This element is of type [MapMarkers](#).

Following is the parent element of the **MapMarkerRule.MapMarkers** element.

Parent elements
<a href="#">MapMarkerRule</a>

The following is the XML Schema definition of the **MapMarkerRule.MapMarkers** element.

```
<xsd:element name="MapMarkers" type="MapMarkersType" minOccurs="1" />
```

## 2.245.11 MapMarkerRule.StartValue

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerRule.StartValue** element specifies the start value for a [MapMarkerRule](#) distribution. The **MapMarkerRule.StartValue** element is optional.

This element is ignored if the value of the [MapColorPaletteRule.DistributionType](#) element is "Custom" or if the value of the [MapColorPaletteRule.DataValue](#) element is not scalar. Otherwise, if the **MapMarkerRule.StartValue** element is not specified, its value MUST be automatically calculated.

Following is the parent element of the **MapMarkerRule.StartValue** element.

Parent elements
MapMarkerRule

The following is the XML Schema definition of the **MapMarkerRule.StartValue** element.

```
<xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
```

## 2.246 MapMarkers

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkers** element specifies a collection of [MapMarker](#) elements. The **MapMarkers** element MUST contain at least one **MapMarker** element.

The following are the parent and child elements of the **MapMarkers** element.

<b>Parent elements</b>
<a href="#">MapMarkerRule</a>

<b>Child elements</b>
<a href="#">MapMarkers.MapMarker</a>

The following is the XML Schema definition of the **MapMarkers** element.

```
<xsd:complexType name="MapMarkersType">
  <xsd:sequence>
    <xsd:element name="MapMarker" type="MapMarkerType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.246.1 MapMarkers.MapMarker

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkers.MapMarker** element specifies a marker to be applied to map points in a [MapPointLayer](#) or in a [MapPolygonLayer](#). The **MapMarkers.MapMarker** element MUST be specified and is of type [MapMarker](#).

Following is the parent element of the **MapMarkers.MapMarker** element.

<b>Parent elements</b>
<a href="#">MapMarkers</a>

The following is the XML Schema definition of the **MapMarkers.MapMarker** element.

```
<xsd:element name="MapMarker" type="MapMarkerType" minOccurs="1"
  maxOccurs="unbounded" />
```

### 2.247 MapMarker

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarker** element specifies a **map marker** to be applied to the center point of a [MapPoint](#) or a [MapPolygon](#).

The following are the parent elements and child element of the **MapMarker** element.

<b>Parent elements</b>
<a href="#">MapMarkerTemplate</a>
<a href="#">MapMarkers</a>

<b>Child elements</b>
<a href="#">MapMarker.MapMarkerImage</a>
<a href="#">MapMarker.MapMarkerStyle</a>

The following is the XML Schema definition of the **MapMarker** element.

```
<xsd:complexType name="MapMarkerType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapMarkerStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapMarkerImage" type="MapMarkerImageType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.247.1 MapMarker.MapMarkerImage

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarker.MapMarkerImage** element specifies an image to be applied to a [MapMarker](#). This element is optional.

This element MUST be specified if the value of the [MapMarker.MapMarkerStyle](#) element is set to "Image". Otherwise, the value of the **MapMarker.MapMarkerImage** element is ignored. This element is of type [MapMarkerImage](#).

Following is the parent element of the **MapMarker.MapMarkerImage** element.

<b>Parent elements</b>
<a href="#">MapMarker</a>

The following is the XML Schema definition of the **MapMarker.MapMarkerImage** element.

```
<xsd:element name="MapMarkerImage" type="MapMarkerImageType" minOccurs="0" />
```

### 2.247.2 MapMarker.MapMarkerStyle

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarker.MapMarkerStyle** element specifies the style for a [MapMarker](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**None:** No marker is used.

**Rectangle:** A rectangle-shaped marker is used.

**Circle:** A circle-shaped marker is used.

**Diamond:** A diamond-shaped marker is used.

**Triangle:** A triangle-shaped marker is used.

**Trapezoid:** A trapezoid-shaped marker is used.



**Star:** A star-shaped marker is used.

**Wedge:** A wedge-shaped marker is used.

**Pentagon:** A pentagon-shaped marker is used.

**PushPin:** A pushpin-shaped marker is used.

**Image:** An image marker is used.

If this element is not present, its value is interpreted as "None".

Following is the parent element of the **MapMarker.MapMarkerStyle** element.

Parent elements
MapMarker

The following is the XML Schema definition of the **MapMarker.MapMarkerStyle** element.

```
<xsd:element name="MapMarkerStyle" type="xsd:string" minOccurs="0" />
```

## 2.248 MapMarkerImage

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapMarkerImage** element specifies an image to be applied to a [MapMarker](#).

The following are the parent and child elements of the **MapMarkerImage** element.

Parent elements
MapMarker

Child elements
<a href="#">MapMarkerImage.MIMETYPE</a>
<a href="#">MapMarkerImage.ResizeMode</a>
<a href="#">MapMarkerImage.Source</a>
<a href="#">MapMarkerImage.TransparentColor</a>
<a href="#">MapMarkerImage.Value</a>

The following is the XML Schema definition of the **MapMarkerImage** element.

```
<xsd:complexType name="MapMarkerImageType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Source" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
</xsd:complexType>
```

```
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.248.1 MapMarkerImage.MIMETYPE

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerImage.MIMETYPE** element specifies the image format for a [MapMarkerImage](#) element. The **MapMarkerImage.MIMETYPE** element is optional. If this element is present, its value MUST be a [ReportMIMETYPE](#).

If the peer element [MapMarkerImage.Source](#) is set to a value other than "Database", the **MapMarkerImage.MIMETYPE** element is ignored.

Following is the parent element of the **MapMarkerImage.MIMETYPE** element.

Parent elements
MapMarkerImage

The following is the XML Schema definition of the **MapMarkerImage.MIMETYPE** element.

```
<xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
```

### 2.248.2 MapMarkerImage.ResizeMode

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerImage.ResizeMode** element specifies the resize mode for the parent [MapMarkerImage](#) element. The **MapMarkerImage.ResizeMode** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**AutoFit:** The image is sized to fit the marker size.

**None:** The image is drawn using its original size.

If this element is not present, its value is interpreted as "AutoFit".

Following is the parent element of the **MapMarkerImage.ResizeMode** element.

Parent elements
MapMarkerImage

The following is the XML Schema definition of the **MapMarkerImage.ResizeMode** element.

```
<xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
```

### 2.248.3 MapMarkerImage.Source

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerImage.Source** element specifies the type of source that is associated with a [MapMarkerImage](#). The **MapMarkerImage.Source** element MUST be specified.

The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**External:** Specifies that the peer [MapMarkerImage.Value](#) element contains a string constant or an expression that evaluates to the location of an image.

**Embedded:** Specifies that the peer **MapMarkerImage.Value** element contains a string constant or an expression that evaluates to the name of an [EmbeddedImage](#) instance in the report.

**Database:** Specifies that the peer **MapMarkerImage.Value** element contains an expression (for example, a field in the database) that evaluates to the binary data for an image.

Following is the parent element of the **MapMarkerImage.Source** element.

Parent elements
MapMarkerImage

The following is the XML Schema definition of the **MapMarkerImage.Source** element.

```
<xsd:element name="Source" type="xsd:string" minOccurs="0" />
```

#### 2.248.4 MapMarkerImage.TransparentColor

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerImage.TransparentColor** element specifies the color to treat as transparent in the [MapMarkerImage](#). The **MapMarkerImage.TransparentColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. Otherwise, the image is displayed as-is.

Following is the parent element of the **MapMarkerImage.TransparentColor** element.

Parent elements
MapMarkerImage

The following is the XML Schema definition of the **MapMarkerImage.TransparentColor** element.

```
<xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
```

#### 2.248.5 MapMarkerImage.Value

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMarkerImage.Value** element depends on the peer [MapMarkerImage.Source](#) element. The **MapMarkerImage.Value** element is optional.

If the value of the peer **MapMarkerImage.Source** element is "External" and if the **MapMarkerImage.Value** element is present, then the value of the **MapMarkerImage.Value** element MUST be a string constant value or an expression that evaluates to the location of an image. Such a location MUST be a [ReportPath](#) or [RdlURL](#) value.

If the value of the peer **MapMarkerImage.Source** element is "Embedded" and if the **MapMarkerImage.Value** element is present, then the value of the **MapMarkerImage.Value** element MUST be a string constant value or an expression that evaluates to the name of an [EmbeddedImage](#) instance in the report.

If the value of the peer **MapMarkerImage.Source** element is "Database" and if the **MapMarkerImage.Value** element is present, then the value of the **MapMarkerImage.Value** element MUST be a string constant value or an expression that evaluates to the binary data for an image.

If this element is not present or has an empty value, the image MUST NOT be displayed.

Following is the parent element of the **MapMarkerImage.Value** element.

Parent elements
<a href="#">MapMarkerImage</a>

The following is the XML Schema definition of the **MapMarkerImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="0" />
```

## 2.249 MapPoints

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPoints** element specifies a collection of [MapPoint](#) elements in a [MapPointLayer](#).

The following are the parent and child elements of the **MapPoints** element.

Parent elements
MapPointLayer

Child elements
<a href="#">MapPoints.MapPoint</a>

The following is the XML Schema definition of the **MapPoints** element.

```
<xsd:complexType name="MapPointsType">
  <xsd:sequence>
    <xsd:element name="MapPoint" type="MapPointType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.249.1 MapPoints.MapPoint

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPoints.MapPoint** element specifies a point-based spatial element to be embedded in a [MapPointLayer](#). This element MUST be specified. This element is of type [MapPoint](#).

Following is the parent element of the **MapPoints.MapPoint** element.

Parent elements
<a href="#">MapPoints</a>

The following is the XML Schema definition of the **MapPoints.MapPoint** element.

```
<xsd:element name="MapPoint" type="MapPointType" minOccurs="1"
             maxOccurs="unbounded" />
```

## 2.250 MapPoint

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPoint** element specifies a point-based spatial element to be embedded in a [MapLineLayer](#).

The following are the parent and child elements of the **MapPoint** element.

Parent elements
<a href="#">MapPoints</a>

Child elements
<a href="#">MapPoint.MapFields</a>
<a href="#">MapPoint.MapMarkerTemplate</a>
<a href="#">MapPoint.UseCustomPointTemplate</a>
<a href="#">MapPoint.VectorData</a>

The following is the XML Schema definition of the **MapPoint** element.

```
<xsd:complexType name="MapPointType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSpatialElement Start-->
    <xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
    <xsd:element name="MapFields" type="MapFieldsType" minOccurs="0" />
    <!--MapSpatialElement End-->
    <xsd:element name="UseCustomPointTemplate" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType"
                minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.250.1 MapPoint.MapFields

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPoint.MapFields** element specifies a collection of [MapField](#) elements to be associated with a **MapPoint**. The **MapPoint.MapFields** element is optional. This element is of type [MapFields](#).

Following is the parent element of the **MapPoint.MapFields** element.

Parent elements
<a href="#">MapPoint</a>

The following is the XML Schema definition of the **MapPoint.MapFields** element.

```
<xsd:element name="MapFields" type="MapFieldsType" minOccurs="0" />
```

## 2.250.2 MapPoint.MapMarkerTemplate

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapPoint.MapMarkerTemplate** element specifies a custom template to be applied to a [MapPoint](#). This element is optional.

This element is ignored if the value of the [MapPoint.UseCustomPointTemplate](#) element is false. The **MapPoint.MapMarkerTemplate** element SHOULD be specified if the value of the **MapPoint.UseCustomPointTemplate** element is true. The **MapPoint.MapMarkerTemplate** element is of type [MapMarkerTemplate](#).

Following is the parent element of the **MapPoint.MapMarkerTemplate** element.

Parent elements
<a href="#">MapPoint</a>

The following is the XML Schema definition of the **MapPoint.MapMarkerTemplate** element.

```
<xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType"
  minOccurs="0" />
```

## 2.250.3 MapPoint.UseCustomPointTemplate

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapPoint.UseCustomPointTemplate** element specifies whether the [MapPoint.MapMarkerTemplate](#) element will be applied to the parent [MapPoint](#) element. The **MapPoint.UseCustomPointTemplate** element is optional.

If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapPoint.UseCustomPointTemplate** element.

Parent elements
<a href="#">MapPoint</a>

The following is the XML Schema definition of the **MapPoint.UseCustomPointTemplate** element.

```
<xsd:element name="UseCustomPointTemplate" type="xsd:string" minOccurs="0" />
```

## 2.250.4 MapPoint.VectorData

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPoint.VectorData** element specifies the vector data for a MapPoint. The **MapPoint.VectorData** element MUST be specified. This element MUST be a base64-encoded string of **Well-Known Binary [ISO19125-2]** format.

Following is the parent element of the **MapPoint.VectorData** element.

Parent elements
<a href="#">MapPoint</a>

The following is the XML Schema definition of the **MapPoint.VectorData** element.

```
<xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
```

## 2.251 MapPolygonLayer

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonLayer** element specifies a point-based layer to be drawn in a [Map](#).

The following are the parent element, attribute, and child elements of the **MapPolygonLayer** element.

Parent elements
<a href="#">MapLayers</a>

Attributes
<a href="#">MapPolygonLayer.Name</a>

Child elements
<a href="#">MapPolygonLayer.DataElementName</a>
<a href="#">MapPolygonLayer.DataElementOutput</a>
<a href="#">MapPolygonLayer.MapBindingFieldPairs</a>
<a href="#">MapPolygonLayer.MapCenterPointRules</a>
<a href="#">MapPolygonLayer.MapMarkerTemplate</a>
<a href="#">MapPolygonLayer.MapDataRegionName</a>

Child elements
<a href="#">MapPolygonLayer.MapFieldDefinitions</a>
<a href="#">MapPolygonLayer.MapPolygonRules</a>
<a href="#">MapPolygonLayer.MapPolygons</a>
<a href="#">MapPolygonLayer.MapPolygonTemplate</a>
<a href="#">MapPolygonLayer.MapShapefile</a>
<a href="#">MapPolygonLayer.MapSpatialDataSet</a>
<a href="#">MapPolygonLayer.MapSpatialDataRegion</a>
<a href="#">MapPolygonLayer.MaximumZoom</a>
<a href="#">MapPolygonLayer.MinimumZoom</a>
<a href="#">MapPolygonLayer.Transparency</a>
<a href="#">MapPolygonLayer.VisibilityMode</a>

The following is the XML Schema definition of the **MapPolygonLayer** element.

```

<xsd:complexType name="MapPolygonLayerType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapLayerStart-->
    <xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <!--MapLayerEnd-->
    <!--MapVectorLayerStart-->
    <xsd:element name="MapDataRegionName" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType"
      minOccurs="0" />
    <xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType"
      minOccurs="0" />
    <xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
    <xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType"
      minOccurs="0" />
    <xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType"
      minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapVectorLayerEnd-->
    <xsd:element name="MapPolygonTemplate" type="MapPolygonTemplateType"
      minOccurs="0" />
    <xsd:element name="MapPolygonRules" type="MapPolygonRulesType"
      minOccurs="0" />
    <xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType"
      minOccurs="0" />
    <xsd:element name="MapCenterPointRules" type="MapPointRulesType"
      minOccurs="0" />
    <xsd:element name="MapPolygons" type="MapPolygonsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>

```



```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.251.1 MapPolygonLayer.Name

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonLayer.Name** attribute specifies a name for the [MapPolygonLayer](#) element. The **MapPolygonLayer.Name** attribute MUST be specified. The value of this attribute MUST be a case-sensitive [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is a **CLS-compliant identifier** [UTR15].

Following is the parent element of the **MapPolygonLayer.Name** attribute.

Parent elements
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.251.2 MapPolygonLayer.DataElementName

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonLayer.DataElementName** element specifies the name for the [MapPolygonLayer](#) to use for a data element or attribute. This element is optional.

If this element is not present, its value is interpreted as the **Name** attribute of the map line layer. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is a **CLS-compliant identifier** [UTR15].

Following is the parent element of the **MapPolygonLayer.DataElementName** element.

Parent elements
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

### 2.251.3 MapPolygonLayer.DataElementOutput

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonLayer.DataElementOutput** element specifies whether this [MapPolygonLayer](#) element will appear in a **data rendering**. This element is optional.

If this element is present, its value MUST be one of the following:

**Output:** Specifies that the item appears in the data rendering output.

**NoOutput:** Specifies that the item does not appear in the data rendering output.

If this element is not present, its value is interpreted as "Output".

Following is the parent element of the **MapPolygonLayer.DataElementOutput** element.

Parent elements
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.251.4 MapPolygonLayer.MapBindingFieldPairs

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonLayer.MapBindingFieldPairs** element specifies a collection of [MapBindingFieldPair](#) elements to be used to bind the map items in the parent [MapPolygonLayer](#) to the data region that is associated with the layer. The **MapPolygonLayer.MapBindingFieldPairs** element is optional.

This element MUST be specified if the value of the [MapPolygonLayer.MapDataRegionName](#) element is specified and the value of the [MapPolygonLayer.MapSpatialDataRegion](#) element is not specified. Otherwise, the **MapPolygonLayer.MapBindingFieldPairs** element is ignored. This element is of type **MapBindingFieldPair**.

Following is the parent element of the **MapPolygonLayer.MapBindingFieldPairs** element.

Parent elements
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.MapBindingFieldPairs** element.

```
<xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType"
  minOccurs="0" />
```

## 2.251.5 MapPolygonLayer.MapCenterPointRules

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonLayer.MapCenterPointRules** element specifies a set of rules to be applied to a [MapPolygon](#) center point in the parent layer. This element is optional. This element is of type [MapPointRules](#).

Following is the parent element of the **MapPolygonLayer.MapCenterPointRules** element.

<b>Parent elements</b>
------------------------

<a href="#">MapPolygonLayer</a>
---------------------------------

The following is the XML Schema definition of the **MapPolygonLayer.MapCenterPointRules** element.

```
<xsd:element name="MapCenterPointRules" type="MapPointRulesType" minOccurs="0" />
```

## 2.251.6 MapPolygonLayer.MapMarkerTemplate

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonLayer.MapMarkerTemplate** element specifies a template for the [MapPolygon](#) center point to be applied to points in the parent **map layer**. This element is optional. This element is of type [MapMarkerTemplate](#).

Following is the parent element of the **MapPolygonLayer.MapMarkerTemplate** element.

<b>Parent elements</b>
------------------------

<a href="#">MapPolygonLayer</a>
---------------------------------

The following is the XML Schema definition of the **MapPolygonLayer.MapMarkerTemplate** element.

```
<xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType" minOccurs="0" />
```

## 2.251.7 MapPolygonLayer.MapDataRegionName

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonLayer.MapDataRegionName** element specifies the data region from which to consume data. This element is optional.

This element **MUST** be specified if the value of the [MapPolygonLayer.MapSpatialDataRegion](#) element is specified.

Following is the parent element of the **MapPolygonLayer.MapDataRegionName** element.

<b>Parent elements</b>
------------------------

<a href="#">MapPolygonLayer</a>
---------------------------------

The following is the XML Schema definition of the **MapPolygonLayer.MapDataRegionName** element.

```
<xsd:element name="MapDataRegionName" type="xsd:string" minOccurs="0" />
```

## 2.251.8 MapPolygonLayer.MapFieldDefinitions

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonLayer.MapFieldDefinitions** element specifies a collection of [MapFieldDefinition](#) elements in the parent [MapPolygonLayer](#). This collection describes the metadata for the fields that are associated with the **map polygons** (specified by [MapPolygon](#) elements). The **MapPolygonLayer.MapFieldDefinitions** element is optional.

This element is ignored if the value of the **MapPolygonLayer.MapPoints** element is not specified. The **MapPolygonLayer.MapFieldDefinitions** element is of type [MapFieldDefinitions](#).

Following is the parent element of the **MapPolygonLayer.MapFieldDefinitions** element.

Parent elements
<a href="#">MapPolygonLayer</a>

The following is the XML Schema definition of the **MapPolygonLayer.MapFieldDefinitions** element.

```
<xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType"
  minOccurs="0" />
```

## 2.251.9 MapPolygonLayer.MapPolygonRules

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonLayer.MapPolygonRules** element specifies a set of rules to be applied to the **map polygons** (specified by [MapPolygon](#) elements) in the parent layer. This element is optional. This element is of type [MapPolygonRules](#).

Following is the parent element of the **MapPolygonLayer.MapPolygonRules** element.

Parent elements
<a href="#">MapPolygonLayer</a>

The following is the XML Schema definition of the **MapPolygonLayer.MapPolygonRules** element.

```
<xsd:element name="MapPolygonRules" type="MapPolygonRulesType" minOccurs="0" />
```

## 2.251.10 MapPolygonLayer.MapPolygons

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonLayer.MapPolygons** element specifies a collection of embedded **map polygons**. This element is optional. This element is of type [MapPolygons](#).

Following is the parent element of the **MapPolygonLayer.MapPolygons** element.

Parent elements
<a href="#">MapPolygonLayer</a>

The following is the XML Schema definition of the **MapPolygonLayer.MapPolygons** element.

```
<xsd:element name="MapPolygons" type="MapPolygonsType" minOccurs="0" />
```

## 2.251.11 MapPolygonLayer.MapPolygonTemplate

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonLayer.MapPolygonTemplate** element specifies a **polygon template** to apply to **map polygons** (specified by [MapPolygon](#) elements) in the parent **map layer**. This element is optional. This element is of type [MapPolygonTemplate](#).

Following is the parent element of the **MapPolygonLayer.MapPolygonTemplate** element.

Parent elements
<a href="#">MapPolygonLayer</a>

The following is the XML Schema definition of the **MapPolygonLayer.MapPolygonTemplate** element.

```
<xsd:element name="MapPolygonTemplate" type="MapPolygonTemplateType"
  minOccurs="0" />
```

## 2.251.12 MapPolygonLayer.MapSpatialDataRegion

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonLayer.MapSpatialDataRegion** element specifies a map data region as the source for the **map polygons** (specified by [MapPolygon](#) elements) for the parent layer. This element is optional. This element is ignored if the value of the [MapPolygonLayer.MapPolygons](#) element is specified.

The **MapPolygonLayer.MapSpatialDataRegion** cannot be present if either of the following elements is present:

- [MapPolygonLayer.MapSpatialDataSet](#)
- [MapPolygonLayer.MapShapefile](#)

The **MapPolygonLayer.MapSpatialDataRegion** element is of type [MapSpatialDataRegion](#).

Following is the parent element of the **MapPolygonLayer.MapSpatialDataRegion** element.

Parent elements
<a href="#">MapPolygonLayer</a>

The following is the XML Schema definition of the **MapPolygonLayer.MapSpatialDataRegion** element.

```
<xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType"
  minOccurs="0" />
```

## 2.251.13 MapPolygonLayer.MapShapefile

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonLayer.MapShapefile** element specifies a **shapefile** as the source for the **map polygons** (specified by [MapPolygon](#) elements) for the parent layer. This element is optional. This element is ignored if the value of the [MapPolygonLayer.MapPolygons](#) element is specified.

The **MapPolygonLayer.MapShapefile** element cannot be present if either of the following elements is present:

- [MapPolygonLayer.MapSpatialDataSet](#)
- [MapPolygonLayer.MapSpatialDataRegion](#)

The **MapPolygonLayer.MapShapefile** element is of type [MapShapefile](#).

Following is the parent element of the **MapPolygonLayer.MapShapefile** element.

Parent elements
<a href="#">MapPolygonLayer</a>

The following is the XML Schema definition of the **MapPolygonLayer.MapShapefile** element.

```
<xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
```

## 2.251.14 MapPolygonLayer.MapSpatialDataSet

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonLayer.MapSpatialDataSet** element specifies a SQL Server dataset as the source for the **map polygons** (specified by [MapPolygon](#) elements) for the parent layer. This element is optional. This element is ignored if the value of the [MapPolygonLayer.MapPolygons](#) element is specified.

The **MapPolygonLayer.MapSpatialDataSet** element cannot be present if either of the following elements is present:

- [MapPolygonLayer.MapSpatialDataRegion](#)
- [MapPolygonLayer.MapShapefile](#)

The **MapPolygonLayer.MapSpatialDataSet** element is of type [MapSpatialDataSet](#).

Following is the parent element of the **MapPolygonLayer.MapSpatialDataSet** element.

Parent elements
<a href="#">MapPolygonLayer</a>

The following is the XML Schema definition of the **MapPolygonLayer.MapSpatialDataSet** element.

```
<xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType" minOccurs="0" />
```

## 2.251.15 MapPolygonLayer.MaximumZoom

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonLayer.MaximumZoom** element specifies the maximum zoom level at which the parent layer is visible. This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 200. This element is ignored if the value of the **MapPolygonLayer.VisibilityMode** element is not "ZoomBased".

Following is the parent element of the **MapPolygonLayer.MaximumZoom** element.

Parent elements
<a href="#">MapPolygonLayer</a>

The following is the XML Schema definition of the **MapPolygonLayer.MaximumZoom** element.

```
<xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
```

## 2.251.16 MapPolygonLayer.MinimumZoom

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonLayer.MinimumZoom** element specifies the minimum zoom level at which the parent layer is visible. This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 50. This element is ignored if the value of the **MapPolygonLayer.VisibilityMode** element is not "ZoomBased".

Following is the parent element of the **MapPolygonLayer.MinimumZoom** element.

Parent elements
<a href="#">MapPolygonLayer</a>

The following is the XML Schema definition of the **MapPolygonLayer.MinimumZoom** element.

```
<xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
```

## 2.251.17 MapPolygonLayer.Transparency

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonLayer.Transparency** element specifies the transparency of the **map polygons** (specified by [MapPolygon](#) elements) in the parent layer as a percentage. This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 100.

Following is the parent element of the **MapPolygonLayer.Transparency** element.

<b>Parent elements</b>
------------------------

<a href="#">MapPolygonLayer</a>
---------------------------------

The following is the XML Schema definition of the **MapPolygonLayer.Transparency** element.

```
<xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
```

## 2.251.18 MapPolygonLayer.VisibilityMode

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonLayer.VisibilityMode** element specifies the visibility mode for a [MapPolygonLayer](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Visible:** The layer is always visible.

**Hidden:** The layer is always hidden.

**ZoomBased:** The layer is shown in a zoom level range that is defined by **MapPolygonLayer.MaximumZoom** and **MapPolygonLayer.MinimumZoom**.

If this element is not present, its value is interpreted as "Visible".

Following is the parent element of the **MapPolygonLayer.VisibilityMode** element.

<b>Parent elements</b>
------------------------

<a href="#">MapPolygonLayer</a>
---------------------------------

The following is the XML Schema definition of the **MapPolygonLayer.VisibilityMode** element.

```
<xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
```

## 2.252 MapPolygonRules

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonRules** element specifies a set of rules to be applied to **map polygons** (specified by [MapPolygon](#) elements) in the parent layer.

The following are the parent and child elements of the **MapPolygonRules** element.

<b>Parent elements</b>
------------------------

<a href="#">MapPolygonLayer</a>
---------------------------------



Child elements
<a href="#">MapLineRules.MapColorPaletteRule</a>
<a href="#">MapLineRules.MapColorRangeRule</a>
<a href="#">MapLineRules.MapCustomColorRule</a>

The following is the XML Schema definition of the **MapPolygonRules** element.

```
<xsd:complexType name="MapPolygonRulesType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType"
      minOccurs="0" />
    <xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType"
      minOccurs="0" />
    <xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType"
      minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.252.1 MapPolygonRules.MapColorPaletteRule

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonRules.MapColorPaletteRule** element specifies a palette of colors for map polygons (specified by [MapPolygon](#) elements) in the parent layer based on the value of the [MapColorPaletteRule.DataValue](#) element. The **MapPolygonRules.MapColorPaletteRule** element is optional. If this element is present, it overrides the [Style.Color](#) grandchild element of the [MapPolygonTemplate](#) element.

The **MapPolygonRules.MapColorPaletteRule** element cannot be present if the either of the following elements is present:

- [MapPolygonRules.MapColorRangeRule](#)
- [MapPolygonRules.MapCustomColorRule](#)

The **MapPolygonRules.MapColorPaletteRule** element is of type [MapColorPaletteRule](#).

Following is the parent element of the **MapPolygonRules.MapColorPaletteRule** element.

Parent elements
<a href="#">MapPolygonRules</a>

The following is the XML Schema definition of the **MapPolygonRules.MapColorPaletteRule** element.

```
<xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType"
  minOccurs="0" />
```

### 2.252.2 MapPolygonRules.MapColorRangeRule

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonRules.MapColorRangeRule** element specifies a color range to be used for coloring map polygons (specified by [MapPolygon](#) elements) in the parent layer based on the value of the [MapColorRangeRule.DataValue](#) element. The **MapPolygonRules.MapColorRangeRule** element is optional. If this element is present, it overrides [MapPolygonTemplate.Style](#).

The **MapPolygonRules.MapColorRangeRule** element cannot be present if either of the following elements is present:

- [MapPolygonRules.MapColorPaletteRule](#)
- [MapPolygonRules.MapCustomColorRule](#)

The **MapPolygonRules.MapColorRangeRule** element is of type [MapColorRangeRule](#).

Following is the parent element of the **MapPolygonRules.MapColorRangeRule** element.

Parent elements
<a href="#">MapPolygonRules</a>

The following is the XML Schema definition of the **MapPolygonRules.MapColorRangeRule** element.

```
<xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType"
  minOccurs="0" />
```

### 2.252.3 MapPolygonRules.MapCustomColorRule

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonRules.MapCustomColorRule** element specifies a user-defined set of colors for map polygons (specified by [MapPolygon](#) elements) in the parent layer based on the value of the [MapCustomColorRule.DataValue](#) element. The **MapPolygonRules.MapCustomColorRule** element is optional. If this element is present, it overrides **MapPolygonTemplate.Style.Color**.

The **MapPolygonRules.MapCustomColorRule** cannot be present if either of the following elements is present:

- [MapPolygonRules.MapColorPaletteRule](#)
- [MapPolygonRules.MapColorRangeRule](#)

The **MapPolygonRules.MapCustomColorRule** element is of type [MapCustomColorRule](#).

Following is the parent element of the **MapPolygonRules.MapCustomColorRule** element.

Parent elements
<a href="#">MapPolygonRules</a>

The following is the XML Schema definition of the **MapPolygonRules.MapCustomColorRule** element.

```
<xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType"
  minOccurs="0" />
```

## 2.253 MapPolygons

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygons** element specifies a collection of [MapPolygon](#) elements in a [MapPolygonLayer](#).

The following are the parent and child element of the **MapPolygons** element.

Parent elements
<a href="#">MapPolygonLayer</a>

Child elements
<a href="#">MapPolygons.MapPolygon</a>

The following is the XML Schema definition of the **MapPolygons** element.

```
<xsd:complexType name="MapPolygonsType">
  <xsd:sequence>
    <xsd:element name="MapPolygon" type="MapPolygonType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.253.1 MapPolygons.MapPolygon

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygons.MapPolygon** element specifies a polygon-based spatial element to be embedded in a [MapPolygonLayer](#). This element **MUST** be specified. This element is of type [MapPolygon](#).

Following is the parent element of the **MapPolygons.MapPolygon** element.

Parent elements
<a href="#">MapPolygons</a>

The following is the XML Schema definition of the **MapPolygons.MapPolygon** element.

```
<xsd:element name="MapPolygon" type="MapPolygonType" minOccurs="1"
  maxOccurs="unbounded" />
```

## 2.254 MapPolygon

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygon** element specifies a polygon-based spatial element (for example, a continent, a country, or a state) to be embedded in a [MapPolygonLayer](#).

The following are the parent and child elements of the **MapPolygon** element.

<b>Parent elements</b>
<a href="#">MapPolygons</a>

<b>Child elements</b>
<a href="#">MapPolygon.MapMarkerTemplate</a>
<a href="#">MapPolygon.MapFields</a>
<a href="#">MapPolygon.MapPolygonTemplate</a>
<a href="#">MapPolygon.UseCustomCenterPointTemplate</a>
<a href="#">MapPolygon.UseCustomPolygonTemplate</a>
<a href="#">MapPolygon.VectorData</a>

The following is the XML Schema definition of the **MapPolygon** element.

```
<xsd:complexType name="MapPolygonType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSpatialElement Start-->
    <xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
    <xsd:element name="MapFields" type="MapFieldsType" minOccurs="0" />
    <!--MapSpatialElement End-->
    <xsd:element name="UseCustomPolygonTemplate" type="xsd:string"
      minOccurs="0" />
    <xsd:element name="MapPolygonTemplate" type="MapPolygonTemplateType"
      minOccurs="0" />
    <xsd:element name="UseCustomCenterPointTemplate" type="xsd:string"
      minOccurs="0" />
    <xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType"
      minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.254.1 MapPolygon.MapMarkerTemplate

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygon.MapMarkerTemplate** element specifies a custom template to be applied to the [MapPolygon](#) center point. This element is optional.

This element is ignored if the value of the [MapPolygon.UseCustomCenterPointTemplate](#) element is false. The **MapPolygon.MapMarkerTemplate** element SHOULD be specified if the value of the **MapPolygon.UseCustomCenterPointTemplate** element is true. This element is of type [MapMarkerTemplate](#).

Following is the parent element of the **MapPolygon.MapMarkerTemplate** element.

<b>Parent elements</b>
<a href="#">MapPolygon</a>

The following is the XML Schema definition of the **MapPolygon.MapMarkerTemplate** element.

```
<xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType"
  minOccurs="0" />
```

## 2.254.2 MapPolygon.MapFields

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygon.MapFields** element specifies a collection of [MapField](#) elements to be associated with a [MapPolygon](#) element. The **MapPolygon.MapFields** element is optional. This element is of type [MapFields](#).

Following is the parent element of the **MapPolygon.MapFields** element.

Parent elements
MapPolygon

The following is the XML Schema definition of the **MapPolygon.MapFields** element.

```
<xsd:element name="MapFields" type="MapFieldsType" minOccurs="0" />
```

## 2.254.3 MapPolygon.MapPolygonTemplate

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygon.MapPolygonTemplate** element specifies a custom template to be applied to a [MapPolygon](#). This element is optional.

This element is ignored if the value of the [MapPolygon.UseCustomPolygonTemplate](#) element is false. The **MapPolygon.MapPolygonTemplate** element SHOULD be specified if the value of the **MapPolygon.UseCustomPolygonTemplate** element is true. This element is of type [MapPolygonTemplate](#).

Following is the parent element of the **MapPolygon.MapPolygonTemplate** element.

Parent elements
MapPolygon

The following is the XML Schema definition of the **MapPolygon.MapPolygonTemplate** element.

```
<xsd:element name="MapPolygonTemplate" type="MapPolygonTemplateType"
  minOccurs="0" />
```

## 2.254.4 MapPolygon.UseCustomCenterPointTemplate

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygon.UseCustomCenterPointTemplate** element specifies whether the [MapPolygon.MapMarkerTemplate](#) element will be applied to the parent [MapPolygon](#) center point. The **MapPolygon.UseCustomCenterPointTemplate** element is optional.

If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapPolygon.UseCustomCenterPointTemplate** element.

Parent elements
MapPolygon

The following is the XML Schema definition of the **MapPolygon.UseCustomCenterPointTemplate** element.

```
<xsd:element name="UseCustomCenterPointTemplate" type="xsd:string"
  minOccurs="0" />
```

### 2.254.5 MapPolygon.UseCustomPolygonTemplate

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygon.UseCustomPolygonTemplate** element specifies whether the [MapPolygon.MapPolygonTemplate](#) element will be applied to the parent map polygon. The **MapPolygon.UseCustomPolygonTemplate** element is optional.

If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapPolygon.UseCustomPolygonTemplate** element.

Parent elements
<a href="#">MapPolygon</a>

The following is the XML Schema definition of the **MapPolygon.UseCustomPolygonTemplate** element.

```
<xsd:element name="UseCustomPolygonTemplate" type="xsd:string" minOccurs="0" />
```

### 2.254.6 MapPolygon.VectorData

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygon.VectorData** element specifies the vector data for a [MapPolygon](#). This element MUST be specified. The value of this element MUST be a base64-encoded string of **Well-Known Binary** [\[ISO19125-2\]](#) format.

Following is the parent element of the **MapPolygon.VectorData** element.

Parent elements
MapPolygon

The following is the XML Schema definition of the **MapPolygon.VectorData** element.

```
<xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
```

## 2.255 MapPolygonTemplate

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonTemplate** element specifies a template to be applied to **map polygons** (specified by [MapPolygon](#) elements) in a **map polygon layer**.

The following are the parent and child elements of the **MapPolygonTemplate** element.

Parent elements
<a href="#">MapPolygonLayer</a>

Child elements
<a href="#">MapPolygonTemplate.ActionInfo</a>
<a href="#">MapPolygonTemplate.CenterPointOffsetX</a>
<a href="#">MapPolygonTemplate.CenterPointOffsetY</a>
<a href="#">MapPolygonTemplate.DataElementLabel</a>
<a href="#">MapPolygonTemplate.DataElementName</a>
<a href="#">MapPolygonTemplate.DataElementOutput</a>
<a href="#">MapPolygonTemplate.Hidden</a>
<a href="#">MapPolygonTemplate.Label</a>
<a href="#">MapPolygonTemplate.LabelPlacement</a>
<a href="#">MapPolygonTemplate.OffsetX</a>
<a href="#">MapPolygonTemplate.OffsetY</a>
<a href="#">MapPolygonTemplate.ScaleFactor</a>
<a href="#">MapPolygonTemplate.ShowLabel</a>
<a href="#">MapPolygonTemplate.Style</a>
<a href="#">MapPolygonTemplate.ToolTip</a>

The following is the XML Schema definition of the **MapPolygonTemplate** element.

```
<xsd:complexType name="MapPolygonTemplateType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSpatialElementTemplateType Start-->
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetY" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
```

```

        <xsd:enumeration value="Output" />
        <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="DataElementLabel" type="xsd:string" minOccurs="0" />
<!--MapSpatialElementTemplateType End-->
<xsd:element name="ScaleFactor" type="xsd:string" minOccurs="0" />
<xsd:element name="CenterPointOffsetX" type="xsd:string" minOccurs="0" />
<xsd:element name="CenterPointOffsetY" type="xsd:string" minOccurs="0" />
<xsd:element name="ShowLabel" type="xsd:string" minOccurs="0" />
<xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

## 2.255.1 MapPolygonTemplate.ActionInfo

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonTemplate.ActionInfo** element specifies a collection of actions to be assigned to map polygons (specified by [MapPolygon](#) elements) in the parent layer. The **MapPolygonTemplate.ActionInfo** element is optional and is of type [ActionInfo](#).

Following is the parent element of the **MapPolygonTemplate.ActionInfo** element.

Parent elements
<a href="#">MapPolygonTemplate</a>

The following is the XML Schema definition of the **MapPolygonTemplate.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

## 2.255.2 MapPolygonTemplate.CenterPointOffsetX

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonTemplate.CenterPointOffset** element specifies the X offset for the center point of a [MapPolygon](#) to which the parent [MapPolygonTemplate](#) is applied. This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapPolygonTemplate.CenterPointOffset** element.

Parent elements
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.CenterPointOffset** element.

```
<xsd:element name="CenterPointOffsetX" type="xsd:string" minOccurs="0" />
```



### 2.255.3 MapPolygonTemplate.CenterPointOffsetY

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonTemplate.CenterPointOffsetY** element specifies the Y offset of the center point of a [MapPolygon](#) to which the parent [MapPolygonTemplate](#) is applied. This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapPolygonTemplate.CenterPointOffsetY** element.

Parent elements
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.CenterPointOffsetY** element.

```
<xsd:element name="CenterPointOffsetY" type="xsd:string" minOccurs="0" />
```

### 2.255.4 MapPolygonTemplate.DataElementLabel

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonTemplate.DataElementLabel** element specifies the label for the [MapPolygonTemplate](#) to use for the data element or attribute. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as the value of the [MapPolygonTemplate.Label](#) property.

Following is the parent element of the **MapPolygonTemplate.DataElementLabel** element.

Parent elements
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.DataElementLabel** element.

```
<xsd:element name="DataElementLabel" type="xsd:string" minOccurs="0" />
```

### 2.255.5 MapPolygonTemplate.DataElementName

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonTemplate.DataElementName** element specifies a name for a [MapPolygonTemplate](#) to use for the data element or attribute. This element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#). If this element is not present, its value is interpreted as "MapDataRow".

Following is the parent element of the **MapPolygonTemplate.DataElementName** element.

Parent elements
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

## 2.255.6 MapPolygonTemplate.DataElementOutput

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonTemplate.DataElementOutput** element specifies whether a [MapPolygonTemplate](#) will appear in a **data rendering**. This element is optional.

If this element is present, its value MUST be one of the following:

**Output:** Specifies that the item appears in the data rendering output.

**NoOutput:** Specifies that the item does not appear in the data rendering output.

If this element is not present, its value is interpreted as "Output".

Following is the parent element of the **MapPolygonTemplate.DataElementOutput** element.

Parent elements
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.255.7 MapPolygonTemplate.Hidden

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonTemplate.Hidden** element specifies whether map polygons (specified by [MapPolygon](#) elements) to which the **polygon template** is applied are hidden. This element is optional.

If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapPolygonTemplate.Hidden** element.

<b>Parent elements</b>
------------------------

<a href="#">MapPolygonTemplate</a>
------------------------------------

The following is the XML Schema definition of the **MapPolygonTemplate.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

## 2.255.8 MapPolygonTemplate.Label

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonTemplate.Label** element specifies the label text for map polygons (specified by [MapPolygon](#) elements) to which this [MapPolygonTemplate](#) is applied. This element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **MapPolygonTemplate.Label** element.

<b>Parent elements</b>
------------------------

<a href="#">MapPolygonTemplate</a>
------------------------------------

The following is the XML Schema definition of the **MapPolygonTemplate.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

## 2.255.9 MapPolygonTemplate.LabelPlacement

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonTemplate.LabelPlacement** element specifies label placement for the [MapPolygon](#) layer. This element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**MiddleCenter:** The label is positioned in the middle center of the polygon.

**MiddleLeft:** The label is positioned in the middle left of the polygon.

**MiddleRight:** The label is positioned in the middle right of the polygon.

**TopCenter:** The label is positioned in the top center of the polygon.

**TopLeft:** The label is positioned in the top left of the polygon.

**TopRight:** The label is positioned in the top right of the polygon.

**BottomCenter:** The label is positioned in the bottom center of the polygon.

**BottomLeft:** The label is positioned in the bottom left of the polygon.

**BottomRight:** The label is positioned in the bottom right of the polygon.

If this element is not present, its value is interpreted as "MiddleCenter".

Following is the parent element of the **MapPolygonTemplate.LabelPlacement** element.

Parent elements
<a href="#">MapPolygonTemplate</a>

The following is the XML Schema definition of the **MapPolygonTemplate.LabelPlacement** element.

```
<xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
```

### 2.255.10 MapPolygonTemplate.OffsetX

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonTemplate.OffsetX** element specifies the X offset in **map coordinates** for map polygons (specified by [MapPolygon](#) elements) to which this [MapPolygonTemplate](#) is applied. This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapPolygonTemplate.OffsetX** element.

Parent elements
<a href="#">MapPolygonTemplate</a>

The following is the XML Schema definition of the **MapPolygonTemplate.OffsetX** element.

```
<xsd:element name="OffsetX" type="xsd:string" minOccurs="0" />
```

### 2.255.11 MapPolygonTemplate.OffsetY

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapPolygonTemplate.OffsetY** element specifies the Y offset in **map coordinates** for map polygons (specified by [MapPolygon](#) elements) to which this [MapPolygonTemplate](#) is applied. This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapPolygonTemplate.OffsetY** element.

Parent elements
<a href="#">MapPolygonTemplate</a>

The following is the XML Schema definition of the **MapPolygonTemplate.OffsetY** element.

```
<xsd:element name="OffsetY" type="xsd:string" minOccurs="0" />
```

## 2.255.12 MapPolygonTemplate.ScaleFactor

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonTemplate.ScaleFactor** element specifies the scale factor for map polygons (specified by [MapPolygon](#) elements) to which this [MapPolygonTemplate](#) is applied. This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 1.

Following is the parent element of the **MapPolygonTemplate.ScaleFactor** element.

Parent elements
<a href="#">MapPolygonTemplate</a>

The following is the XML Schema definition of the **MapPolygonTemplate.ScaleFactor** element.

```
<xsd:element name="ScaleFactor" type="xsd:string" minOccurs="0" />
```

## 2.255.13 MapPolygonTemplate.ShowLabel

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonTemplate.ShowLabel** element specifies whether labels for a [MapPolygon](#) are displayed. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Auto:** The labels are shown if there is no collision.

**True:** The labels are always shown.

**False:** The labels are hidden.

If this element is not present, its value is interpreted as "Auto".

Following is the parent element of the **MapPolygonTemplate.ShowLabel** element.

Parent elements
<a href="#">MapPolygonTemplate</a>

The following is the XML Schema definition of the **MapPolygonTemplate.ShowLabel** element.

```
<xsd:element name="ShowLabel" type="xsd:string" minOccurs="0" />
```

## 2.255.14 MapPolygonTemplate.Style

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonTemplate.Style** element specifies style information for map polygons (specified by [MapPolygon](#) elements) to which a [MapPolygonTemplate](#) is applied. This element is optional and is of type [Style](#).

Following is the parent element of the **MapPolygonTemplate.Style** element.

Parent elements
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.255.15 MapPolygonTemplate.ToolTip

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapPolygonTemplate.ToolTip** element specifies the tooltip text for the map polygons (specified by [MapPolygon](#) elements) to which a [MapPolygonTemplate](#) is applied. The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **MapPolygonTemplate.ToolTip** element is optional.

If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **MapPolygonTemplate.ToolTip** element.

Parent elements
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

## 2.256 MapTileLayer

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTileLayer** element specifies a raster-based **map layer** to be drawn in a [Map](#).

The following are the parent element, attribute, and child elements of the **MapTileLayer** element.

Parent elements
<a href="#">MapLayers</a>

Attributes
<a href="#">MapTileLayer.Name</a>

Child elements
<a href="#">MapTileLayer.MapTiles</a>
<a href="#">MapTileLayer.MaximumZoom</a>
<a href="#">MapTileLayer.MinimumZoom</a>
<a href="#">MapTileLayer.TileStyle</a>
<a href="#">MapTileLayer.Transparency</a>
<a href="#">MapTileLayer.UseSecureConnection</a>
<a href="#">MapTileLayer.VisibilityMode</a>

The following is the XML Schema definition of the **MapTileLayer** element.

```
<xsd:complexType name="MapTileLayerType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapLayerStart-->
    <xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <!--MapLayerEnd-->
    <xsd:element name="TileStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseSecureConnection" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapTiles" type="MapTilesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.256.1 MapTileLayer.Name

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTileLayer.Name** attribute specifies a name for the [MapTileLayer](#) element. The **MapTileLayer.Name** attribute MUST be specified. The value of this attribute MUST be a case-sensitive [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#).

Following is the parent element of the **MapTileLayer.Name** attribute.

Parent elements
MapTileLayer

The following is the XML Schema definition of the **MapTileLayer.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.256.2 MapTileLayer.MapTiles

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTileLayer.MapTiles** element specifies a collection of [MapTile](#) elements to be embedded in a [MapTileLayer](#) in a [Map](#). This element is optional and is of type [MapTiles](#).

Following is the parent element of the **MapTileLayer.MapTiles** element.

Parent elements
<a href="#">MapTileLayer</a>

The following is the XML Schema definition of the **MapTileLayer.MapTiles** element.

```
<xsd:element name="MapTiles" type="MapTilesType" minOccurs="0" />
```

### 2.256.3 MapTileLayer.MaximumZoom

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapTileLayer.MaximumZoom** element specifies the maximum zoom level at which the parent layer is visible in a [Map](#). This element is optional.

If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 200. This element is ignored if the value of the [MapTileLayer.VisibilityMode](#) element is not "ZoomBased".

Following is the parent element of the **MapTileLayer.MaximumZoom** element.

Parent elements
<a href="#">MapTileLayer</a>

The following is the XML Schema definition of the **MapTileLayer.MaximumZoom** element.

```
<xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
```

### 2.256.4 MapTileLayer.MinimumZoom

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapTileLayer.MinimumZoom** element specifies the minimum zoom level at which the parent layer is visible in a [Map](#). This element is optional.

If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 50. This element is ignored if the value of the [MapTileLayer.VisibilityMode](#) element is not "ZoomBased".

Following is the parent element of the **MapTileLayer.MinimumZoom** element.

Parent elements
<a href="#">MapTileLayer</a>

The following is the XML Schema definition of the **MapTileLayer.MinimumZoom** element.

```
<xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
```



## 2.256.5 MapTileLayer.TileStyle

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapTileLayer.TileStyle** element specifies the style of a [MapTile](#) in a [Map](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Road:** Tiles contain roads and streets.

**Aerial:** Tiles contain aerial imagery of Earth.

**Hybrid:** Tiles contain aerial imagery, borders, and roads.

If this element is not present, its value is interpreted as "Road".

Following is the parent element of the **MapTileLayer.TileStyle** element.

Parent elements
<a href="#">MapTileLayer</a>

The following is the XML Schema definition of the **MapTileLayer.TileStyle** element.

```
<xsd:element name="TileStyle" type="xsd:string" minOccurs="0" />
```

## 2.256.6 MapTileLayer.Transparency

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapTileLayer.Transparency** element specifies the transparency of the **map tiles** (specified by [MapTile](#) elements) in the parent layer as a percentage. This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 100.

Following is the parent element of the **MapTileLayer.Transparency** element.

Parent elements
<a href="#">MapTileLayer</a>

The following is the XML Schema definition of the **MapTileLayer.Transparency** element.

```
<xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
```

## 2.256.7 MapTileLayer.UseSecureConnection

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapTileLayer.UseSecureConnection** element specifies whether to use secure connection to retrieve the tiles for a [MapTileLayer](#). The **MapTileLayer.UseSecureConnection** element is optional.

If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapTileLayer.UseSecureConnection** element.

Parent elements
MapTileLayer

The following is the XML Schema definition of the **MapTileLayer.UseSecureConnection** element.

```
<xsd:element name="UseSecureConnection" type="xsd:string" minOccurs="0" />
```

## 2.256.8 MapTileLayer.VisibilityMode

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapTileLayer.VisibilityMode** element specifies the visibility mode for a [MapTileLayer](#). This element is optional.

If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Visible:** The layer is always visible.

**Hidden:** The layer is always hidden.

**ZoomBased:** The layer is shown in a zoom level range that is defined by the [MapTileLayer.MaximumZoom](#) and [MapTileLayer.MinimumZoom](#) elements.

If the **MapTileLayer.VisibilityMode** element is not present, its value is interpreted as "Visible".

Following is the parent element of the **MapTileLayer.VisibilityMode** element.

Parent elements
MapTileLayer

The following is the XML Schema definition of the **MapTileLayer.VisibilityMode** element.

```
<xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
```

## 2.257 MapTiles

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapTiles** element specifies a collection of [MapTile](#) elements for a [MapTileLayer](#).

The following are the parent and child elements of the **MapTiles** element.

Parent elements
MapTileLayer

<b>Child elements</b>
-----------------------

<a href="#">MapTiles.MapTile</a>
----------------------------------

The following is the XML Schema definition of the **MapTiles** element.

```
<xsd:complexType name="MapTilesType">
  <xsd:sequence>
    <xsd:element name="MapTile" type="MapTileType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.257.1 MapTiles.MapTile

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTiles.MapTile** element specifies a [MapTile](#) image to be embedded in a [MapTileLayer](#). This element **MUST** be specified.

Following is the parent element of the **MapTiles.MapTile** element.

<b>Parent elements</b>
------------------------

<a href="#">MapTiles</a>
--------------------------

The following is the XML Schema definition of the **MapTiles.MapTile** element.

```
<xsd:element name="MapTile" type="MapTileType" minOccurs="1"
  maxOccurs="unbounded" />
```

## 2.258 MapTile

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTile** element specifies a **MapTile** image to be embedded in a [MapTileLayer](#).

The following are the parent elements and child elements of the **MapTile** element.

<b>Parent elements</b>
------------------------

<a href="#">MapTiles</a>
--------------------------

<b>Child elements</b>
-----------------------

<a href="#">MapTile.Name</a>
------------------------------

<a href="#">MapTile.TileData</a>
----------------------------------

The following is the XML Schema definition of the **MapTile** element.

```

<xsd:complexType name="MapTileType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" minOccurs="1" />
    <xsd:element name="TileData" type="xsd:string" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.258.1 MapTile.Name

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTile.Name** element specifies the name for a [MapTile](#). This element MUST be specified. The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Following is the parent element of the **MapTile.Name** element.

Parent elements
MapTile

The following is the XML Schema definition of the **MapTile.Name** element.

```

<xsd:element name="Name" type="xsd:string" minOccurs="1" />

```

### 2.258.2 MapTile.TileData

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTile.TileData** element specifies the image data for a [MapTile](#) in a [Map](#). This element MUST be specified. The value of this element MUST be a base64-encoded string.

Following is the parent element of the **MapTile.TileData** element.

Parent elements
MapTile

The following is the XML Schema definition of the **MapTile.TileData** element.

```

<xsd:element name="TileData" type="xsd:string" minOccurs="1" />

```

### 2.259 MapLegends

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLegends** element specifies a collection of [MapLegend](#) elements to be drawn in a [Map](#). The **MapLegends** element MUST contain at least one **MapLegend** element.

The following are the parent and child elements of the **MapLegends** element.

Parent elements
Map

Child elements
<a href="#">MapLegends.MapLegend</a>

The following is the XML Schema definition of the **MapLegends** element.

```
<xsd:complexType name="MapLegendsType">
  <xsd:sequence>
    <xsd:element name="MapLegend" type="MapLegendType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.259.1 MapLegends.MapLegend

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLegends.MapLegend** element specifies a legend to be drawn in a [Map](#). This element MUST be specified and is of type [MapLegend](#).

Following is the parent element of the **MapLegends.MapLegend** element.

Parent elements
<a href="#">MapLegends</a>

The following is the XML Schema definition of the **MapLegends.MapLegend** element.

```
<xsd:element name="MapLegend" type="MapLegendType" minOccurs="1"
  maxOccurs="unbounded" />
```

### 2.260 MapLegend

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLegend** element specifies a legend to be drawn in a [Map](#).

The following are the parent element, attribute, and child elements of the **MapLegend** element.

Parent elements
<a href="#">MapLegends</a>

Attributes
<a href="#">MapLegend.Name</a>

Child elements
<a href="#">MapLegend.ActionInfo</a>
<a href="#">MapLegend.AutoFitTextDisabled</a>
<a href="#">MapLegend.BottomMargin</a>
<a href="#">MapLegend.DockOutsideViewport</a>
<a href="#">MapLegend.EquallySpacedItems</a>
<a href="#">MapLegend.Hidden</a>
<a href="#">MapLegend.InterlacedRows</a>
<a href="#">MapLegend.InterlacedRowsColor</a>
<a href="#">MapLegend.Layout</a>
<a href="#">MapLegend.LeftMargin</a>
<a href="#">MapLegend.MapLegendTitle</a>
<a href="#">MapLegend.MapLocation</a>
<a href="#">MapLegend.MapSize</a>
<a href="#">MapLegend.MinFontSize</a>
<a href="#">MapLegend.Position</a>
<a href="#">MapLegend.RightMargin</a>
<a href="#">MapLegend.Style</a>
<a href="#">MapLegend.TextWrapThreshold</a>
<a href="#">MapLegend.ToolTip</a>
<a href="#">MapLegend.TopMargin</a>
<a href="#">MapLegend.ZIndex</a>

The following is the XML Schema definition of the **MapLegend** element.

```
<xsd:complexType name="MapLegendType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSubItem Start-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
    <xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <!--MapSubItem End-->
  </xsd:choice>
</xsd:complexType>
```

```

<!--MapDockableSubItem Start-->
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
<xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<!--MapDockableSubItem End-->
<xsd:element name="Layout" type="xsd:string" minOccurs="0" />
<xsd:element name="MapLegendTitle" type="MapLegendTitleType" minOccurs="0" />
<xsd:element name="AutoFitTextDisabled" type="xsd:string" minOccurs="0" />
<xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
<xsd:element name="InterlacedRows" type="xsd:string" minOccurs="0" />
<xsd:element name="InterlacedRowsColor" type="xsd:string" minOccurs="0" />
<xsd:element name="EquallySpacedItems" type="xsd:string" minOccurs="0" />
<xsd:element name="TextWrapThreshold" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.260.1 MapLegend.Name

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapLegend.Name** attribute specifies a name for a [MapLegend](#). This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive [String](#) ([XMLSCHEMA2/21](#) section 3.2.1) that is a **CLS-compliant identifier** [[UTR15](#)].

Following is the parent element of the **MapLegend.Name** attribute.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.260.2 MapLegend.ActionInfo

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapLegend.ActionInfo** element specifies a set of actions for a [MapLegend](#) in a [Map](#). This element is optional and is of type [ActionInfo](#).

Following is the parent element of the **MapLegend.ActionInfo** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.260.3 MapLegend.AutoFitTextDisabled

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLegend.AutoFitTextDisabled** element specifies whether text will be autosized to fit in the [MapLegend](#) area. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapLegend.AutoFitTextDisabled** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.AutoFitTextDisabled** element.

```
<xsd:element name="AutoFitTextDisabled" type="xsd:string" minOccurs="0" />
```

### 2.260.4 MapLegend.BottomMargin

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLegend.BottomMargin** element specifies the bottom margin for a [MapLegend](#) and its parent [Map](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapLegend.BottomMargin** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.BottomMargin** element.

```
<xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
```

### 2.260.5 MapLegend.DockOutsideViewport

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLegend.DockOutsideViewport** element specifies whether the [MapLegend](#) will be docked outside a [MapViewport](#). This element is optional.

If the **MapLegend.DockOutsideViewport** element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false. This element is ignored if the value of the [MapLegend.MapLocation](#) element is defined.

Following is the parent element of the **MapLegend.DockOutsideViewport** element.

Parent elements
MapLegend



The following is the XML Schema definition of the **MapLegend.DockOutsideViewport** element.

```
<xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
```

## 2.260.6 **MapLegend.EquallySpacedItems**

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLegend.EquallySpacedItems** element specifies whether legend items are equally spaced in a [MapLegend](#). This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapLegend.EquallySpacedItems** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.EquallySpacedItems** element.

```
<xsd:element name="EquallySpacedItems" type="xsd:string" minOccurs="0" />
```

## 2.260.7 **MapLegend.Hidden**

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLegend.Hidden** element specifies whether a [MapLegend](#) will be hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapLegend.Hidden** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

## 2.260.8 **MapLegend.InterlacedRows**

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLegend.InterlacedRows** element specifies whether the legend rows in a [MapLegend](#) will use interlaced colors. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapLegend.InterlacedRows** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.InterlacedRows** element.

```
<xsd:element name="InterlacedRows" type="xsd:string" minOccurs="0" />
```

## 2.260.9 MapLegend.InterlacedRowsColor

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLegend.InterlacedRowsColor** element specifies the color of the interlaced rows in a [MapLegend](#). This element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**. If this element is not present, its value is interpreted as "LightGray".

Following is the parent element of the **MapLegend.InterlacedRowsColor** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.InterlacedRowsColor** element.

```
<xsd:element name="InterlacedRowsColor" type="xsd:string" minOccurs="0" />
```

## 2.260.10 MapLegend.Layout

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLegend.Layout** element specifies the arrangement of labels in a [MapLegend](#). This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**AutoTable:** The labels are automatically arranged to fit.

**Column:** The labels are arranged in a column.

**Row:** The labels are arranged in a row.

**WideTable:** The labels are arranged in a wide table.

**TallTable:** The labels are arranged in a tall table.

If this element is not present, its value is interpreted as "AutoTable".

Following is the parent element of the **MapLegend.Layout** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.Layout** element.

```
<xsd:element name="Layout" type="xsd:string" minOccurs="0" />
```

### 2.260.11 MapLegend.LeftMargin

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLegend.LeftMargin** element specifies the left margin for a [MapLegend](#) and its parent [Map](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapLegend.LeftMargin** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.LeftMargin** element.

```
<xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
```

### 2.260.12 MapLegend.MapLegendTitle

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLegend.MapLegendTitle** element specifies a title for a [MapLegend](#). This element is optional and is of type [MapLegendTitle](#).

Following is the parent element of the **MapLegend.MapLegendTitle** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.MapLegendTitle** element.

```
<xsd:element name="MapLegendTitle" type="MapLegendTitleType" minOccurs="0" />
```

### 2.260.13 MapLegend.MapLocation

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLegend.MapLocation** element specifies the location of a [MapLegend](#). This element is optional. If this element is present, [MapLegend.Position](#) is ignored. The **MapLegend.MapLocation** element is of type [MapLocation](#).

Following is the parent element of the **MapLegend.MapLocation** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.MapLocation** element.

```
<xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
```

## 2.260.14 MapLegend.MapSize

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLegend.MapSize** element specifies the size of a [MapLegend](#). This element is optional. If this element is not present, the size of the map legend MUST be automatically calculated. This element is of type [MapSize](#).

Following is the parent element of the **MapLegend.MapSize** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.MapSize** element.

```
<xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
```

## 2.260.15 MapLegend.MinFontSize

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLegend.MinFontSize** element specifies the minimum font size for the auto-sizing algorithm of [MapLegend](#) text. This element is optional.

If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**. If this element is not present, its value is interpreted as 7pt. This element is ignored if the value of the [MapLegend.AutoFitTextDisabled](#) element is true.

Following is the parent element of the **MapLegend.MinFontSize** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.MinFontSize** element.

```
<xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
```

## 2.260.16 MapLegend.Position

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapLegend.Position** element specifies the position of the [MapLegend](#) in a [Map](#). This element is optional. This element is ignored if the value of the [MapLegend.MapLocation](#) element is specified. If the **MapLegend.Position** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**TopCenter:** The legend is positioned at the top center.

**TopLeft:** The legend is positioned at the top left.

**TopRight:** The legend is positioned at the top right.

**LeftTop:** The legend is positioned at the left top.

**LeftCenter:** The legend is positioned at the left center.

**LeftBottom:** The legend is positioned at the left bottom.

**RightTop:** The legend is positioned at the right top.

**RightCenter:** The legend is positioned at the right center.

**RightBottom:** The legend is positioned at the right bottom.

**BottomRight:** The legend is positioned at the bottom right.

**BottomCenter:** The legend is positioned at the bottom center.

**BottomLeft:** The legend is positioned at the bottom left.

If this element is not present, its value is interpreted as "TopCenter".

Following is the parent element of the **MapLegend.Position** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.Position** element.

```
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
```

## 2.260.17 MapLegend.RightMargin

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLegend.RightMargin** element specifies the right margin for a [MapLegend](#) and its parent [Map](#). This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapLegend.RightMargin** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.RightMargin** element.

```
<xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
```

## 2.260.18 MapLegend.Style

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLegend.Style** element specifies style information for a [MapLegend](#). This element is optional and is of type [Style](#).

Following is the parent element of the **MapLegend.Style** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.260.19 MapLegend.TextWrapThreshold

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLegend.TextWrapThreshold** element specifies the number of characters after which to wrap the legend text in a [MapLegend](#). This element is optional.

If this element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2](#) section 3.3.17) or an expression that evaluates to an **Integer**. If this element is not present, its value is interpreted as 25. The value of this element MUST NOT be a negative number.

Following is the parent element of the **MapLegend.TextWrapThreshold** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.TextWrapThreshold** element.

```
<xsd:element name="TextWrapThreshold" type="xsd:string" minOccurs="0" />
```

## 2.260.20 MapLegend.ToolTip

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLegend.ToolTip** element specifies the tooltip text for a [MapLegend](#). The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **MapLegend.ToolTip** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **MapLegend.ToolTip** element.

Parent elements
MapLegend

The following is the XML Schema definition of the **MapLegend.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

## 2.260.21 MapLegend.TopMargin

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLegend.TopMargin** element specifies the top margin for a [MapLegend](#) and its parent [Map](#). The **MapLegend.TopMargin** element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapLegend.TopMargin** element.

Parent elements
<a href="#">MapLegend</a>

The following is the XML Schema definition of the **MapLegend.TopMargin** element.

```
<xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
```

## 2.260.22 MapLegend.ZIndex

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLegend.ZIndex** element specifies the drawing order of a [MapLegend](#) within its container. This element is optional. If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapLegend.ZIndex** element.

Parent elements
<a href="#">MapLegend</a>

The following is the XML Schema definition of the **MapLegend.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
```

## 2.261 MapLegendTitle

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLegendTitle** element specifies the title of a [MapLegend](#).

The following are the parent and child elements of the **MapLegendTitle** element.

Parent elements
<a href="#">MapLegend</a>

Child elements
<a href="#">MapLegendTitle.Caption</a>

Child elements
<a href="#">MapLegendTitle.Style</a>
<a href="#">MapLegendTitle.TitleSeparator</a>
<a href="#">MapLegendTitle.TitleSeparatorColor</a>

The following is the XML Schema definition of the **MapLegendTitle** element.

```
<xsd:complexType name="MapLegendTitleType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Caption" type="xsd:string" minOccurs="0" />
    <xsd:element name="TitleSeparator" type="xsd:string" minOccurs="0" />
    <xsd:element name="TitleSeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.261.1 MapLegendTitle.Caption

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLegendTitle.Caption** element specifies the caption for the title of a map legend. This element is optional. If this element is present, its value **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **MapLegendTitle.Caption** element.

Parent elements
<a href="#">MapLegendTitle</a>

The following is the XML Schema definition of the **MapLegendTitle.Caption** element.

```
<xsd:element name="Caption" type="xsd:string" minOccurs="0" />
```

### 2.261.2 MapLegendTitle.Style

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLegendTitle.Style** element specifies style information for a [MapLegendTitle](#) element. This element is optional and is of type [Style](#).

Following is the parent element of the **MapLegendTitle.Style** element.

Parent elements
MapLegendTitle

The following is the XML Schema definition of the **MapLegendTitle.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```



### 2.261.3 **MapLegendTitle.TitleSeparator**

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLegendTitle.TitleSeparator** element specifies the type of separator to use for a [MapLegendTitle](#) element. The **MapLegendTitle.TitleSeparator** element is optional. The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**None:** No separator is applied.

**Line:** A line separator is applied.

**ThickLine:** A thick line separator is applied.

**DoubleLine:** A double line separator is applied.

**DashLine:** A dashed line separator is applied.

**DotLine:** A dotted line separator is applied.

**GradientLine:** A gradient line separator is applied.

**ThickGradientLine:** A thick gradient line separator is applied.

If this element is not present, its value is interpreted as "None".

Following is the parent element of the **MapLegendTitle.TitleSeparator** element.

Parent elements
MapLegendTitle

The following is the XML Schema definition of the **MapLegendTitle.TitleSeparator** element.

```
<xsd:element name="TitleSeparator" type="xsd:string" minOccurs="0" />
```

### 2.261.4 **MapLegendTitle.TitleSeparatorColor**

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLegendTitle.TitleSeparatorColor** element specifies the title separator color in a [MapLegendTitle](#). This element is optional. If this element is present, its value MUST be an [RdlColor](#) value or an expression that evaluates to an **RdlColor**. If this element is not present, its value is interpreted as "Gray".

Following is the parent element of the **MapLegendTitle.TitleSeparatorColor** element.

Parent elements
MapLegendTitle

The following is the XML Schema definition of the **MapLegendTitle.TitleSeparatorColor** element.

```
<xsd:element name="TitleSeparatorColor" type="xsd:string" minOccurs="0" />
```

## 2.262 MapTitles

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitles** element specifies an ordered list of [MapTitle](#) elements to be drawn in a [Map](#) element. The **MapTitles** element MUST contain at least one **MapTitle** element.

The following are the parent and child elements of the **MapTitles** element.

Parent elements
<a href="#">Map</a>

Child elements
<a href="#">MapTitles.MapTitle</a>

The following is the XML Schema definition of the **MapTitles** element.

```
<xsd:complexType name="MapTitlesType">  
  <xsd:sequence>  
    <xsd:element name="MapTitle" type="MapTitleType" minOccurs="1"  
      maxOccurs="unbounded" />  
  </xsd:sequence>  
  <xsd:anyAttribute namespace="##other" processContents="lax" />  
</xsd:complexType>
```

### 2.262.1 MapTitles.MapTitle

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitles.MapTitle** element specifies the title for a [Map](#) element. The **MapTitles.MapTitle** element MUST be specified at least once. This element is of type [MapTitle](#).

Following is the parent element of the **MapTitles.MapTitle** element.

Parent elements
<a href="#">MapTitles</a>

The following is the XML Schema definition of the **MapTitles.MapTitle** element.

```
<xsd:element name="MapTitle" type="MapTitleType" minOccurs="1"  
  maxOccurs="unbounded" />
```

## 2.263 MapTitle

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitle** element specifies a title to be drawn in a [Map](#) element.

The following are the parent element, attribute, and child elements of the **MapTitle** element.

Parent elements
<a href="#">MapTitles</a>

Attributes
<a href="#">MapTitle.Name</a>

Child elements
<a href="#">MapTitle.ActionInfo</a>
<a href="#">MapTitle.Angle</a>
<a href="#">MapTitle.BottomMargin</a>
<a href="#">MapTitle.DockOutsideViewport</a>
<a href="#">MapTitle.Hidden</a>
<a href="#">MapTitle.LeftMargin</a>
<a href="#">MapTitle.MapLocation</a>
<a href="#">MapTitle.MapSize</a>
<a href="#">MapTitle.Position</a>
<a href="#">MapTitle.RightMargin</a>
<a href="#">MapTitle.Style</a>
<a href="#">MapTitle.Text</a>
<a href="#">MapTitle.TextShadowOffset</a>
<a href="#">MapTitle.ToolTip</a>
<a href="#">MapTitle.TopMargin</a>
<a href="#">MapTitle.ZIndex</a>

The following is the XML Schema definition of the **MapTitle** element.

```
<xsd:complexType name="MapTitleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSubItem Start-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
    <xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

```

<xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
<!--MapSubItem End-->
<!--MapDockableSubItem Start-->
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
<xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<!--MapDockableSubItem End-->
<xsd:element name="Text" type="xsd:string" minOccurs="0" />
<xsd:element name="Angle" type="xsd:string" minOccurs="0" />
<xsd:element name="TextShadowOffset" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.263.1 MapTitle.Name

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapTitle.Name** attribute specifies the name of a [MapTitle](#). The **MapTitle.Name** attribute MUST be specified. The value of this attribute MUST be a case-sensitive [String](#) ([XMLSCHEMA2/21](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#).

Following is the parent element of the **MapTitle.Name** attribute.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.263.2 MapTitle.ActionInfo

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapTitle.ActionInfo** element specifies a set of actions for a [MapTitle](#) element. The **MapTitle.ActionInfo** element is optional and is of type [ActionInfo](#).

Following is the parent element of the **MapTitle.ActionInfo** element.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

### 2.263.3 MapTitle.Angle

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitle.Angle** element specifies the rotation angle of a MapTitle element in degrees. This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not specified, its value is interpreted as 0.

Following is the parent element of the **MapTitle.Angle** element.

Parent elements
<a href="#">MapTitle</a>

The following is the XML Schema definition of the **MapTitle.Angle** element.

```
<xsd:element name="Angle" type="xsd:string" minOccurs="0" />
```

### 2.263.4 MapTitle.BottomMargin

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitle.BottomMargin** element specifies the bottom margin for a [MapTitle](#) element and its parent [Map](#) element. The **MapTitle.BottomMargin** element is optional.

If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapTitle.BottomMargin** element.

Parent elements
<a href="#">MapTitle</a>

The following is the XML Schema definition of the **MapTitle.BottomMargin** element.

```
<xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
```

### 2.263.5 MapTitle.DockOutsideViewport

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitle.DockOutsideViewport** element specifies whether a [MapTitle](#) element will be docked outside a **map viewport**. The **MapTitle.DockOutsideViewport** element is optional.

If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false. This element is ignored if the value of the [MapTitle.MapLocation](#) element is defined.

Following is the parent element of the **MapTitle.DockOutsideViewport** element.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.DockOutsideViewport** element.

```
<xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
```

## 2.263.6 MapTitle.Hidden

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitle.Hidden** element specifies whether a [MapTitle](#) element is hidden. The **MapTitle.Hidden** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapTitle.Hidden** element.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

## 2.263.7 MapTitle.LeftMargin

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitle.LeftMargin** element specifies the left margin for a [MapTitle](#) element and its parent [Map](#) element. The **MapTitle.LeftMargin** element is optional.

If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapTitle.LeftMargin** element.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.LeftMargin** element.

```
<xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
```

## 2.263.8 MapTitle.MapLocation

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitle.MapLocation** element specifies the location of a [MapTitle](#) element. The **MapTitle.MapLocation** element is optional and is of type [MapLocation](#). If the **MapTitle.MapLocation** element is present, the [MapTitle.Position](#) element is ignored.

Following is the parent element of the **MapTitle.MapLocation** element.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.MapLocation** element.

```
<xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
```

## 2.263.9 MapTitle.MapSize

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapTitle.MapSize** element specifies the size of a [MapTitle](#) element. The **MapTitle.MapSize** element is optional. If this element is not present, the map title size MUST be automatically calculated. This element is of type [MapSize](#).

Following is the parent element of the **MapTitle.MapSize** element.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.MapSize** element.

```
<xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
```

## 2.263.10 MapTitle.Position

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapTitle.Position** element specifies the position of a [MapTitle](#) element in a [Map](#) element. The **MapTitle.Position** element is optional.

This element is ignored if the value of the **MapTitle.MapLocation** element is specified. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**TopCenter:** The title is positioned at the top center.

**TopLeft:** The title is positioned at the top left.

**TopRight:** The title is positioned at the top right.

**LeftTop:** The title is positioned at the left top.

**LeftCenter:** The title is positioned at the left center.

**LeftBottom:** The title is positioned at the left bottom.

**RightTop:** The title is positioned at the right top.

**RightCenter:** The title is positioned at the right center.

**RightBottom:** The title is positioned at the right bottom.

**BottomRight:** The title is positioned at the bottom right.

**BottomCenter:** The title is positioned at the bottom center.

**BottomLeft:** The title is positioned at the bottom left.

If this element is not present, its value is interpreted as "TopCenter".

Following is the parent element of the **MapTitle.Position** element.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.Position** element.

```
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
```

### 2.263.11 MapTitle.RightMargin

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitle.RightMargin** element specifies the right margin for a [MapTitle](#) element and its parent [Map](#) element. The **MapTitle.RightMargin** element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapTitle.RightMargin** element.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.RightMargin** element.

```
<xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
```

### 2.263.12 MapTitle.Style

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitle.Style** element specifies style information for a [MapTitle](#) element. The **MapTitle.Style** element is optional and is of type [Style](#).

Following is the parent element of the **MapTitle.Style** element.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.Style** element.



```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

### 2.263.13 MapTitle.Text

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitle.Text** element specifies the text for a [MapTitle](#) element. The **MapTitle.Text** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **MapTitle.Text** element.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.Text** element.

```
<xsd:element name="Text" type="xsd:string" minOccurs="0" />
```

### 2.263.14 MapTitle.TextShadowOffset

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitle.TextShadowOffset** element specifies the size of the text shadow of a [MapTitle](#) element. The **MapTitle.TextShadowOffset** element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapTitle.TextShadowOffset** element.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.TextShadowOffset** element.

```
<xsd:element name="TextShadowOffset" type="xsd:string" minOccurs="0" />
```

### 2.263.15 MapTitle.ToolTip

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitle.ToolTip** element specifies the tooltip text for a [MapTitle](#) element. The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **MapTitle.ToolTip** element is optional. If the **MapTitle.ToolTip** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as an empty string.

Following is the parent element of the **MapTitle.ToolTip** element.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.ToolTip** element.

```
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
```

## 2.263.16 MapTitle.TopMargin

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitle.TopMargin** element specifies the top margin for a [MapTitle](#) element and its parent [Map](#) element. This element is optional.

If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapTitle.TopMargin** element.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.TopMargin** element.

```
<xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
```

## 2.263.17 MapTitle.ZIndex

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapTitle.ZIndex** element specifies the drawing order of a [MapTitle](#) element. The **MapTitle.ZIndex** element is optional.

If this element is present, its value MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapTitle.ZIndex** element.

Parent elements
MapTitle

The following is the XML Schema definition of the **MapTitle.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
```

## 2.264 MapViewport

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapViewport** element specifies a **map viewport**.

The following are the parent and child elements of the **MapViewport** element.

Parent elements
<a href="#">Map</a>

Child elements
<a href="#">MapViewport.BottomMargin</a>
<a href="#">MapViewport.ContentMargin</a>
<a href="#">MapViewport.GridUnderContent</a>
<a href="#">MapViewport.LeftMargin</a>
<a href="#">MapViewport.MapCoordinateSystem</a>
<a href="#">MapViewport.MapLimits</a>
<a href="#">MapViewport.MapLocation</a>
<a href="#">MapViewport.MapMeridians</a>
<a href="#">MapViewport.MapParallels</a>
<a href="#">MapViewport.MapProjection</a>
<a href="#">MapViewport.MapSize</a>
<a href="#">MapViewport.MapCustomView</a>
<a href="#">MapViewport.MapDataBoundView</a>
<a href="#">MapViewport.MapElementView</a>
<a href="#">MapViewport.MaximumZoom</a>
<a href="#">MapViewport.MinimumZoom</a>
<a href="#">MapViewport.ProjectionCenterX</a>
<a href="#">MapViewport.ProjectionCenterY</a>
<a href="#">MapViewport.RightMargin</a>
<a href="#">MapViewport.SimplificationResolution</a>
<a href="#">MapViewport.Style</a>
<a href="#">MapViewport.TopMargin</a>
MapViewport.ZIndex

The following is the XML Schema definition of the **MapViewport** element.

```
<xsd:complexType name="MapViewportType">  
  <xsd:choice maxOccurs="unbounded">  
    <!--MapSubItem Start-->
```

```

<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
<xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
<xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
<!--MapSubItem End-->
<xsd:element name="MapCoordinateSystem" type="xsd:string" minOccurs="0" />
<xsd:element name="MapProjection" type="xsd:string" minOccurs="0" />
<xsd:element name="ProjectionCenterX" type="xsd:string" minOccurs="0" />
<xsd:element name="ProjectionCenterY" type="xsd:string" minOccurs="0" />
<xsd:element name="MapCustomView" type="MapCustomViewType" minOccurs="0" />
<xsd:element name="MapElementView" type="MapElementViewType" minOccurs="0" />
<xsd:element name="MapDataBoundView" type="MapDataBoundViewType" minOccurs="0" />
<xsd:element name="MapLimits" type="MapLimitsType" minOccurs="0" />
<xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
<xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
<xsd:element name="SimplificationResolution" type="xsd:string" minOccurs="0" />

<xsd:element name="ContentMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="MapMeridians" type="MapGridLinesType" minOccurs="0" />
<xsd:element name="MapParallels" type="MapGridLinesType" minOccurs="0" />
<xsd:element name="GridUnderContent" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

## 2.264.1 MapViewport.BottomMargin

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapViewport.BottomMargin** element specifies the bottom margin for a [MapViewport](#) element and its parent [Map](#) element. The **MapViewport.BottomMargin** element is optional.

If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapViewport.BottomMargin** element.

Parent elements
MapViewport

The following is the XML Schema definition of the **MapViewport.BottomMargin** element.

```

<xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />

```

## 2.264.2 MapViewport.ContentMargin

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapViewport.ContentMargin** element specifies the margin between a [MapViewport](#) element and its content. The **MapViewport.ContentMargin** element is optional.

If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 10pt.

Following is the parent element of the **MapViewport.ContentMargin** element.

Parent elements
<a href="#">MapViewport</a>

The following is the XML Schema definition of the **MapViewport.ContentMargin** element.

```
<xsd:element name="ContentMargin" type="xsd:string" minOccurs="0" />
```

### 2.264.3 MapViewport.GridUnderContent

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.GridUnderContent** element specifies whether the **map meridians** and **map parallels** will be drawn under the map content. This element is optional.

If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapViewport.GridUnderContent** element.

Parent elements
<a href="#">MapViewport</a>

The following is the XML Schema definition of the **MapViewport.GridUnderContent** element.

```
<xsd:element name="GridUnderContent" type="xsd:string" minOccurs="0" />
```

### 2.264.4 MapViewport.LeftMargin

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.LeftMargin** element specifies the left margin for a [MapViewport](#) element and its parent [Map](#) element. The **MapViewport.LeftMargin** element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapViewport.LeftMargin** element.

Parent elements
<a href="#">MapViewport</a>

The following is the XML Schema definition of the **MapViewport.LeftMargin** element.

```
<xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
```

## 2.264.5 MapViewport.MapCoordinateSystem

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.MapCoordinateSystem** element specifies the map coordinate system for a [Map](#) element. The **MapViewport.MapCoordinateSystem** element is optional.

If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Planar:** The planar coordinate system is used.

**Geographic:** The geographic coordinate system is used.

If this element is not present, its value is interpreted as "Planar".

Following is the parent element of the **MapViewport.MapCoordinateSystem** element.

Parent elements
<a href="#">MapViewport</a>

The following is the XML Schema definition of the **MapViewport.MapCoordinateSystem** element.

```
<xsd:element name="MapCoordinateSystem" type="xsd:string" minOccurs="0" />
```

## 2.264.6 MapViewport.MapLimits

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.MapLimits** element specifies limits for a **map viewport**. The **MapViewport.MapLimits** element is optional. If this element is not present, it is interpreted as automatically calculated. This element is of type [MapLimits](#).

Following is the parent element of the **MapViewport.MapLimits** element.

Parent elements
<a href="#">MapViewport</a>

The following is the XML Schema definition of the **MapViewport.MapLimits** element.

```
<xsd:element name="MapLimits" type="MapLimitsType" minOccurs="0" />
```

## 2.264.7 MapViewport.MapLocation

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.MapLocation** element specifies the location of a [MapViewport](#) element. The **MapViewport.MapLocation** element is optional and is of type [MapLocation](#).

Following is the parent element of the **MapViewport.MapLocation** element.

Parent elements
MapViewport

The following is the XML Schema definition of the **MapViewport.MapLocation** element.

```
<xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
```

## 2.264.8 MapViewport.MapMeridians

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.MapMeridians** element specifies the horizontal grid for a [Map](#) element. The **MapViewport.MapMeridians** element is optional and is of type [MapMeridians](#).

Following is the parent element of the **MapViewport.MapMeridians** element.

Parent elements
<a href="#">MapViewport</a>

The following is the XML Schema definition of the **MapViewport.MapMeridians** element.

```
<xsd:element name="MapMeridians" type="MapGridLinesType" minOccurs="0" />
```

## 2.264.9 MapViewport.MapParallels

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.MapParallels** element specifies the vertical grid for a [Map](#) element. The **MapViewport.MapParallels** element is optional and is of type [MapParallels](#).

Following is the parent element of the **MapViewport.MapParallels** element.

Parent elements
<a href="#">MapViewport</a>

The following is the XML Schema definition of the **MapViewport.MapParallels** element.

```
<xsd:element name="MapParallels" type="MapGridLinesType" minOccurs="0" />
```

## 2.264.10 MapViewport.MapProjection

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.MapProjection** element specifies the projection to be used for a [Map](#) element. The **MapViewport.MapProjection** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Equiangular:** Equiangular projection is used.

**Mercator:** Mercator projection is used.

**Robinson:** Robinson projection is used.

**Fahey:** Fahey projection is used.

**Eckert1:** Eckert1 projection is used.

**Eckert3:** Eckert3 projection is used.

**HammerAitoff:** HammerAitoff projection is used.

**Wagner3:** Wagner3 projection is used.

**Bonne:** Bonne projection is used.

If this element is not present, its value is interpreted as "Equirectangular". This element is ignored if the value of the [MapViewport.MapCoordinateSystem](#) element is "Planar".

Following is the parent element of the **MapViewport.MapProjection** element.

Parent elements
<a href="#">MapViewport</a>

The following is the XML Schema definition of the **MapViewport.MapProjection** element.

```
<xsd:element name="MapProjection" type="xsd:string" minOccurs="0" />
```

## 2.264.11 MapViewport.MapSize

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.MapSize** element specifies the size of a [MapViewport](#) element. The **MapViewport.MapSize** element is optional. If this element is not present, the **map viewport** size MUST be automatically calculated. This element is of type [MapSize](#).

Following is the parent element of the **MapViewport.MapSize** element.

Parent elements
<a href="#">MapViewport</a>

The following is the XML Schema definition of the **MapViewport.MapSize** element.

```
<xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
```

## 2.264.12 MapViewport.MapCustomView

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.MapCustomView** element specifies the center of a [MapViewport](#) element based on an arbitrary point. The **MapViewport.MapCustomView** element is optional and is of type [MapCustomView](#).

Following is the parent element of the **MapViewport.MapCustomView** element.



Parent elements
MapViewport

The following is the XML Schema definition of the **MapViewport.MapCustomView** element.

```
<xsd:element name="MapCustomView" type="MapCustomViewType" minOccurs="0" />
```

## 2.264.13 MapViewport.MapDataBoundView

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.MapDataBoundView** element specifies the center of a [MapViewport](#) element based on the map items that are bound to data. The **MapViewport.MapDataBoundView** element is optional and is of type [MapDataBoundView](#).

Following is the parent element of the **MapViewport.MapDataBoundView** element.

Parent elements
MapViewport

The following is the XML Schema definition of the **MapViewport.MapDataBoundView** element.

```
<xsd:element name="MapDataBoundView" type="MapDataBoundViewType" minOccurs="0" />
```

## 2.264.14 MapViewport.MapElementView

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.MapElementView** element specifies the center of a [MapViewport](#) element based on a map spatial element. The **MapViewport.MapElementView** element is optional and is of type [MapElementView](#).

Following is the parent element of the **MapViewport.MapElementView** element.

Parent elements
MapViewport

The following is the XML Schema definition of the **MapViewport.MapElementView** element.

```
<xsd:element name="MapElementView" type="MapElementViewType" minOccurs="0" />
```

## 2.264.15 MapViewport.MaximumZoom

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.MaximumZoom** element specifies the maximum zoom level that is allowed in a [Map](#) element. The **MapViewport.MaximumZoom** element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 20000.

Following is the parent element of the **MapViewport.MaximumZoom** element.

Parent elements
<a href="#">MapViewport</a>

The following is the XML Schema definition of the **MapViewport.MaximumZoom** element.

```
<xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
```

## 2.264.16 MapViewport.MinimumZoom

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.MinimumZoom** element specifies the minimum zoom level that is allowed in a [Map](#) element. The **MapViewport.MinimumZoom** element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 20.

Following is the parent element of the **MapViewport.MinimumZoom** element.

Parent elements
<a href="#">MapViewport</a>

The following is the XML Schema definition of the **MapViewport.MinimumZoom** element.

```
<xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
```

## 2.264.17 MapViewport.ProjectionCenterX

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.ProjectionCenterX** element specifies the X projection center of a [MapViewport](#) element in map coordinates. The **MapViewport.ProjectionCenterX** element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value MUST be automatically calculated.

Following is the parent element of the **MapViewport.ProjectionCenterX** element.

Parent elements
MapViewport

The following is the XML Schema definition of the **MapViewport.ProjectionCenterX** element.

```
<xsd:element name="ProjectionCenterX" type="xsd:string" minOccurs="0" />
```

## 2.264.18 MapViewport.ProjectionCenterY

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.ProjectionCenterY** element specifies the Y projection center of a [MapViewport](#) element in map coordinates. The **MapViewport.ProjectionCenterY** element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value MUST be automatically calculated.

Following is the parent element of the **MapViewport.ProjectionCenterY** element.

Parent elements
MapViewport

The following is the XML Schema definition of the **MapViewport.ProjectionCenterY** element.

```
<xsd:element name="ProjectionCenterY" type="xsd:string" minOccurs="0" />
```

## 2.264.19 MapViewport.RightMargin

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.RightMargin** element specifies the right margin for a [MapViewport](#) element and its parent [Map](#) element. The **MapViewport.RightMargin** element is optional.

If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapViewport.RightMargin** element.

Parent elements
MapViewport

The following is the XML Schema definition of the **MapViewport.RightMargin** element.

```
<xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
```

## 2.264.20 MapViewport.SimplificationResolution

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapViewport.SimplificationResolution** element specifies the resolution that is used to simplify map polygons and map lines. The resolution is specified in map coordinates. This element is optional.

If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapViewport.SimplificationResolution** element.

Parent elements
-----------------

<a href="#">MapViewport</a>
-----------------------------

The following is the XML Schema definition of the **MapViewport.SimplificationResolution** element.

```
<xsd:element name="SimplificationResolution" type="xsd:string" minOccurs="0" />
```

## 2.264.21 MapViewport.Style

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapViewport.Style** element specifies style information for a [MapViewport](#) element. The **MapViewport.Style** element is optional and is of type [Style](#).

Following is the parent element of the **MapViewport.Style** element.

Parent elements
-----------------

<a href="#">MapViewport</a>
-----------------------------

The following is the XML Schema definition of the **MapViewport.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.264.22 MapViewport.TopMargin

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapViewport.TopMargin** element specifies the top margin for a [MapViewport](#) element and its parent [Map](#) element. This element is optional.

If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapViewport.TopMargin** element.

Parent elements
-----------------

<a href="#">MapViewport</a>
-----------------------------

The following is the XML Schema definition of the **MapViewport.TopMargin** element.

```
<xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
```

## 2.264.23 MapViewport.ZIndex

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapViewport.ZIndex** element specifies the drawing order of a [MapViewport](#) element within its container. This element is optional. If this element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) or an expression that evaluates to an **Integer**, and its value MUST NOT be negative. If this element is not present, its value is interpreted as 0.

Following is the parent element of the **MapViewport.ZIndex** element.

Parent elements
<a href="#">MapViewport</a>

The following is the XML Schema definition of the **MapViewport.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
```

## 2.265 MapCustomView

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapCustomView** element specifies the center of a [MapViewport](#) element based on an arbitrary point.

The following are the parent and child elements of the **MapCustomView** element.

Parent elements
<a href="#">MapViewport</a>

Child elements
<a href="#">MapCustomView.CenterX</a>
<a href="#">MapCustomView.CenterY</a>
<a href="#">MapCustomView.Zoom</a>

The following is the XML Schema definition of the **MapCustomView** element.

```
<xsd:complexType name="MapCustomViewType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Zoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="CenterX" type="xsd:string" minOccurs="0" />
    <xsd:element name="CenterY" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.265.1 MapCustomView.CenterX

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapCustomView.CenterX** element specifies the x-coordinate of the [MapViewport](#) element center as a percentage of the map content. The **MapCustomView.CenterX** element is optional. If

this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 50.

Following is the parent element of the **MapCustomView.CenterX** element.

Parent elements
<a href="#">MapCustomView</a>

The following is the XML Schema definition of the **MapCustomView.CenterX** element.

```
<xsd:element name="CenterX" type="xsd:string" minOccurs="0" />
```

## 2.265.2 MapCustomView.CenterY

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapCustomView.CenterY** element specifies the y-coordinate of the [MapViewport](#) element center as a percentage of the map content. The **MapCustomView.CenterY** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 50.

Following is the parent element of the **MapCustomView.CenterY** element.

Parent elements
<a href="#">MapCustomView</a>

The following is the XML Schema definition of the **MapCustomView.CenterY** element.

```
<xsd:element name="CenterY" type="xsd:string" minOccurs="0" />
```

## 2.265.3 MapCustomView.Zoom

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapCustomView.Zoom** element specifies the zoom level of a [MapViewport](#) element. The **MapCustomView.Zoom** element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 100.

Following is the parent element of the **MapCustomView.Zoom** element.

Parent elements
<a href="#">MapCustomView</a>

The following is the XML Schema definition of the **MapCustomView.Zoom** element.

```
<xsd:element name="Zoom" type="xsd:string" minOccurs="0" />
```

## 2.266 MapDataBoundView

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapDataBoundView** element specifies the center of a [MapViewport](#) element based on the **map items** that are bound to data.

The following are the parent and child elements of the **MapDataBoundView** element.

Parent elements
<a href="#">MapViewport</a>

Child elements
<a href="#">MapCustomView.Zoom</a>

The following is the XML Schema definition of the **MapDataBoundView** element.

```
<xsd:complexType name="MapCustomViewType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Zoom" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.266.1 MapDataBoundView.Zoom

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapDataBoundView.Zoom** element specifies the zoom level of a [MapViewport](#) element. The **MapDataBoundView.Zoom** element is optional. If this element is present, its value **MUST** be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 100.

Following is the parent element of the **MapDataBoundView.Zoom** element.

Parent elements
<a href="#">MapDataBoundView</a>

The following is the XML Schema definition of the **MapDataBoundView.Zoom** element.

```
<xsd:element name="Zoom" type="xsd:string" minOccurs="0" />
```

### 2.267 MapElementView

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapElementView** element specifies the center of a [MapViewport](#) element based on a **map spatial element**.

The following are the parent and child elements of the **MapElementView** element.

Parent elements
<a href="#">MapViewport</a>

Child elements
<a href="#">MapElementView.LayerName</a>
<a href="#">MapElementView.MapBindingFieldPairs</a>
<a href="#">MapCustomView.Zoom</a>

The following is the XML Schema definition of the **MapElementView** element.

```
<xsd:complexType name="MapElementViewType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Zoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="LayerName" type="xsd:string" minOccurs="1" />
    <xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType"
      minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.267.1 MapElementView.LayerName

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapElementView.LayerName** element specifies the **map layer** name that contains the spatial element that is to be centered in a [MapViewport](#) element. The **MapElementView.LayerName** element **MUST** be specified.

Following is the parent element of the **MapElementView.LayerName** element.

Parent elements
<a href="#">MapElementView</a>

The following is the XML Schema definition of the **MapElementView.LayerName** element.

```
<xsd:element name="LayerName" type="xsd:string" minOccurs="1" />
```

### 2.267.2 MapElementView.MapBindingFieldPairs

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapElementView.MapBindingFieldPairs** element specifies a collection of [MapBindingFieldPair](#) elements that are used to identify the spatial element that is to be centered in a [MapViewport](#) element. The **MapElementView.MapBindingFieldPairs** element is optional.

If this element is not present, its value is interpreted as centering the viewport on the bounding rectangle of all the spatial elements that are contained in the layer that is specified by [MapElementView.LayerName](#). The **MapElementView.MapBindingFieldPairs** element is of type [MapBindingFieldPairs](#).

Following is the parent element of the **MapElementView.MapBindingFieldPairs** element.



Parent elements
-----------------

<a href="#">MapElementView</a>
--------------------------------

The following is the XML Schema definition of the **MapElementView.MapBindingFieldPairs** element.

```
<xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType" minOccurs="0" />
```

### 2.267.3 MapElementView.Zoom

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapElementView.Zoom** element specifies the zoom level of a [MapViewport](#) element. The **MapElementView.Zoom** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value is interpreted as 100.

Following is the parent element of the **MapElementView.Zoom** element.

Parent elements
-----------------

<a href="#">MapElementView</a>
--------------------------------

The following is the XML Schema definition of the **MapElementView.Zoom** element.

```
<xsd:element name="Zoom" type="xsd:string" minOccurs="0" />
```

### 2.268 MapLimits

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **MapLimits** element specifies the limits of a [MapViewport](#) element.

The following are the parent and child elements of the **MapLimits** element.

Parent elements
-----------------

MapViewport
-------------

Child elements
----------------

<a href="#">MapLimits.MaximumX</a>
------------------------------------

<a href="#">MapLimits.MaximumY</a>
------------------------------------

<a href="#">MapLimits.MinimumX</a>
------------------------------------

<a href="#">MapLimits.MinimumY</a>
------------------------------------

The following is the XML Schema definition of the **MapLimits** element.

```
<xsd:complexType name="MapLimitsType">
```

```

<xsd:choice maxOccurs="unbounded">
  <xsd:element name="MinimumX" type="xsd:string" minOccurs="0" />
  <xsd:element name="MinimumY" type="xsd:string" minOccurs="0" />
  <xsd:element name="MaximumX" type="xsd:string" minOccurs="0" />
  <xsd:element name="MaximumY" type="xsd:string" minOccurs="0" />
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.268.1 MapLimits.MaximumX

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLimits.MaximumX** element specifies the maximum X value in **map coordinates** for a [MapViewport](#) element. The **MapLimits.MaximumX** element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value MUST be automatically calculated.

Following is the parent element of the **MapLimits.MaximumX** element.

Parent elements
<a href="#">MapLimits</a>

The following is the XML Schema definition of the **MapLimits.MaximumX** element.

```
<xsd:element name="MaximumX" type="xsd:string" minOccurs="0" />
```

### 2.268.2 MapLimits.MaximumY

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLimits.MaximumY** element specifies the maximum Y value in **map coordinates** for a [MapViewport](#) element. This element is optional. If this element is present, its value MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value MUST be automatically calculated.

Following is the parent element of the **MapLimits.MaximumY** element.

Parent elements
<a href="#">MapLimits</a>

The following is the XML Schema definition of the **MapLimits.MaximumY** element.

```
<xsd:element name="MaximumY" type="xsd:string" minOccurs="0" />
```

### 2.268.3 MapLimits.MinimumX

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLimits.MinimumX** element specifies the minimum X value in **map coordinates** for a [MapViewport](#) element. This element is optional. If this element is present, its value MUST be a [Float](#)

([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value **MUST** be automatically calculated.

Following is the parent element of the **MapLimits.MinimumX** element.

Parent elements
<a href="#">MapLimits</a>

The following is the XML Schema definition of the **MapLimits.MinimumX** element.

```
<xsd:element name="MinimumX" type="xsd:string" minOccurs="0" />
```

## 2.268.4 MapLimits.MinimumY

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapLimits.MinimumY** element specifies the minimum Y value in **map coordinates** for a [MapViewport](#) element. The **MapLimits.MinimumY** element is optional. If this element is present, its value **MUST** be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value **MUST** be automatically calculated.

Following is the parent element of the **MapLimits.MinimumY** element.

Parent element
<a href="#">MapLimits</a>

The following is the XML Schema definition of the **MapLimits.MinimumY** element.

```
<xsd:element name="MinimumY" type="xsd:string" minOccurs="0" />
```

## 2.269 MapMeridians

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMeridians** element specifies the horizontal grid for a [Map](#) element.

The following are the parent and child elements of the **MapMeridians** element.

Parent elements
<a href="#">MapViewport</a>

Child elements
<a href="#">MapMeridians.Hidden</a>
<a href="#">MapMeridians.Interval</a>
<a href="#">MapMeridians.LabelPosition</a>
<a href="#">MapMeridians.ShowLabels</a>

<b>Child elements</b>
-----------------------

<a href="#">MapMeridians.Style</a>
------------------------------------

The following is the XML Schema definition of the **MapMeridians** element.

```
<xsd:complexType name="MapGridLinesType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelPosition" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.269.1 MapMeridians.Hidden

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMeridians.Hidden** element specifies whether the **map meridians** are hidden. This element is optional. If this element is present, its value **MUST** be a [Boolean](#) ([XMLSCHEMA2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapMeridians.Hidden** element.

<b>Parent elements</b>
------------------------

<a href="#">MapMeridians</a>
------------------------------

The following is the XML Schema definition of the **MapMeridians.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

### 2.269.2 MapMeridians.Interval

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMeridians.Interval** element specifies the interval between the **map meridians**. This element is optional. If this element is present, its value **MUST** be a [Float](#) ([XMLSCHEMA2] section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value **MUST** be automatically calculated.

Following is the parent element of the **MapMeridians.Interval** element.

<b>Parent elements</b>
------------------------

<a href="#">MapMeridians</a>
------------------------------

The following is the XML Schema definition of the **MapMeridians.Interval** element.

```
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
```

### 2.269.3 MapMeridians.LabelPosition

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMeridians.LabelPosition** element specifies the label position for **map meridians**. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Near:** The labels are positioned at the top of the meridians.

**OneQuarter:** The labels are positioned between the top and center of the meridians.

**Center:** The labels are positioned at the center of the meridians.

**ThreeQuarters:** The labels are positioned between the center and bottom of the meridians.

**Far:** The labels are positioned at the bottom of the meridians.

If this element is not present, its value is interpreted as "Near".

Following is the parent element of the **MapMeridians.LabelPosition** element.

Parent elements
<a href="#">MapMeridians</a>

The following is the XML Schema definition of the **MapMeridians.LabelPosition** element.

```
<xsd:element name="LabelPosition" type="xsd:string" minOccurs="0" />
```

### 2.269.4 MapMeridians.ShowLabels

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMeridians.ShowLabels** element specifies whether labels for the **map meridians** are hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapMeridians.ShowLabels** element.

Parent elements
<a href="#">MapMeridians</a>

The following is the XML Schema definition of the **MapMeridians.ShowLabels** element.

```
<xsd:element name="ShowLabels" type="xsd:string" minOccurs="0" />
```

### 2.269.5 MapMeridians.Style

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapMeridians.Style** element specifies style information for **map meridians**. This element is optional and is of type [Style](#).

Following is the parent element of the **MapMeridians.Style** element.

Parent elements
<a href="#">MapMeridians</a>

The following is the XML Schema definition of the **MapMeridians.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.270 MapParallels

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapParallels** element specifies the vertical grid of a [Map](#).

The following are the parent and child elements of the **MapParallels** element.

Parent elements
<a href="#">MapViewport</a>

Child elements
<a href="#">MapParallels.Hidden</a>
<a href="#">MapParallels.Interval</a>
<a href="#">MapParallels.LabelPosition</a>
<a href="#">MapParallels.ShowLabels</a>
<a href="#">MapParallels.Style</a>

The following is the XML Schema definition of the **MapParallels** element.

```
<xsd:complexType name="MapGridLinesType">  
  <xsd:choice maxOccurs="unbounded">  
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />  
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />  
    <xsd:element name="ShowLabels" type="xsd:string" minOccurs="0" />  
    <xsd:element name="LabelPosition" type="xsd:string" minOccurs="0" />  
    <xsd:element name="Style" type="StyleType" minOccurs="0" />  
    <xsd:any namespace="##other" processContents="lax" />  
  </xsd:choice>  
  <xsd:anyAttribute namespace="##other" processContents="lax" />  
</xsd:complexType>
```

### 2.270.1 MapParallels.Hidden

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapParallels.Hidden** element specifies whether **map parallels** are hidden. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or

an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapParallels.Hidden** element.

Parent elements
<a href="#">MapParallels</a>

The following is the XML Schema definition of the **MapParallels.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

## 2.270.2 MapParallels.Interval

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapParallels.Interval** element specifies the interval between **map parallels**. This element is optional. If this element is present, its value MUST be a [Float](#) ([XMLSCHEMA2](#) section 3.2.4) or an expression that evaluates to a **Float**. If this element is not present, its value MUST be automatically calculated.

Following is the parent element of the **MapParallels.Interval** element.

Parent elements
<a href="#">MapParallels</a>

The following is the XML Schema definition of the **MapParallels.Interval** element.

```
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
```

## 2.270.3 MapParallels.LabelPosition

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **MapParallels.LabelPosition** element specifies the label position for the **map parallels**. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/21](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Near:** The labels are positioned to the left of the parallels.

**OneQuarter:** The labels are positioned between the left and center of the parallels.

**Center:** The labels are positioned at the center of the parallels.

**ThreeQuarters:** The labels are positioned between the center and right of the parallels.

**Far:** The labels are positioned to the right of the parallels.

If this element is not present, its value is interpreted as "Near".

Following is the parent element of the **MapParallels.LabelPosition** element.

<b>Parent elements</b>
<a href="#">MapParallels</a>

The following is the XML Schema definition of the **MapParallels.LabelPosition** element.

```
<xsd:element name="LabelPosition" type="xsd:string" minOccurs="0" />
```

## 2.270.4 MapParallels.ShowLabels

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapParallels.ShowLabels** element specifies whether the labels for **map parallels** are hidden. This element is optional.

If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **MapParallels.ShowLabels** element.

<b>Parent elements</b>
<a href="#">MapParallels</a>

The following is the XML Schema definition of the **MapParallels.ShowLabels** element.

```
<xsd:element name="ShowLabels" type="xsd:string" minOccurs="0" />
```

## 2.270.5 MapParallels.Style

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The **MapParallels.Style** element specifies style information for **map parallels**. This element is optional and is of type [Style](#).

Following is the parent element of the **MapParallels.Style** element.

<b>Parent elements</b>
<a href="#">MapParallels</a>

The following is the XML Schema definition of the **MapParallels.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.271 CustomData

*Applies to [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomData** parent element defines the data and data manipulation for a [CustomReportItem](#). The data is represented as columns and rows that can optionally be filtered, sorted, grouped, nested,



and repeated. Custom report items that have a **CustomData** element child are considered to be data regions.

The quantity of [DataRow](#) elements within the **CustomReportItem** MUST equal the quantity of [DataMember](#) elements that have a [DataRowHierarchy](#) element ancestor and no **DataMember** element descendent.

The quantity of [DataCell](#) element children in each **DataRow** element within the **CustomReportItem** MUST equal the quantity of **DataMember** elements that have a [DataColumnHierarchy](#) element ancestor and no **DataMember** element descendant.

Following are the parent and child elements of the **CustomData** element.

Parent elements
CustomReportItem

Child elements
<a href="#">CustomData.DataColumnHierarchy</a>
<a href="#">CustomData.DataRowHierarchy</a>
<a href="#">CustomData.DataRows</a>
<a href="#">CustomData.DataSetName</a>
<a href="#">CustomData.Filters</a>
<a href="#">CustomData.SortExpressions</a>
<a href="#">CustomData.DataColumnGroupings</a>
<a href="#">CustomData.DataRowGroupings</a>

**Applies to [RDL 2011/01](#)**

Child elements
<a href="#">CustomData.Relationship</a>

The following is the XML Schema definition of the **CustomData** element in RDL 2005/01.

```
<xsd:complexType name="CustomDataType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="DataColumnGroupings" type="DataColumnGroupingsType"
      minOccurs="0" />
    <xsd:element name="DataRowGroupings" type="DataRowGroupingsType"
      minOccurs="0" />
    <xsd:element name="DataRows" type="DataRowsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **CustomData** element in RDL 2008/01.

```

<xsd:complexType name="CustomDataType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType"
      minOccurs="0" />
    <xsd:element name="DataColumnHierarchy" type="DataColumnHierarchyType"
      minOccurs="0" />
    <xsd:element name="DataRowHierarchy" type="DataRowHierarchyType"
      minOccurs="0" />
    <xsd:element name="DataRows" type="DataRowsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **CustomData** element in RDL 2010/01 and RDL 2016/01.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, 5.6, and 5.7, respectively. For more information about macro- and micro-versioned schemas, see section 2.1.

```

<xsd:complexType name="CustomDataType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType"
      minOccurs="0" />
    <xsd:element name="DataColumnHierarchy" type="DataColumnHierarchyType"
      minOccurs="0" />
    <xsd:element name="DataRowHierarchy" type="DataRowHierarchyType"
      minOccurs="0" />
    <xsd:element name="DataRows" type="DataRowsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.271.1 CustomData.DataColumnHierarchy

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomData.DataColumnHierarchy** element specifies the column hierarchy of data for a [CustomReportItem](#). This element **MUST** be specified exactly once and is of type [DataColumnHierarchy](#).

Following is the parent element of the **CustomData.DataColumnHierarchy** element.

Parent elements
<a href="#">CustomData</a>

The following is the XML Schema definition of the **CustomData.DataColumnHierarchy** element.

```

<xsd:element name="DataColumnHierarchy" type="DataColumnHierarchyType"
  minOccurs="0" />

```

## 2.271.2 CustomData.DataRowHierarchy

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomData.DataRowHierarchy** element specifies the row hierarchy of data for a [CustomReportItem](#). This element MUST be specified exactly once and is of type [DataRowHierarchy](#).

Following is the parent element of the **CustomData.DataRowHierarchy** element.

Parent elements
<a href="#">CustomData</a>

The following is the XML Schema definition of the **CustomData.DataRowHierarchy** element.

```
<xsd:element name="DataRowHierarchy" type="DataRowHierarchyType" minOccurs="0" />
```

## 2.271.3 CustomData.DataRows

Applies to [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomData.DataRows** element specifies the rows of data for a [CustomReportItem](#). This element MUST be specified exactly once and is of type [DataRows](#).

Following is the parent element of the **CustomData.DataRows** element.

Parent elements
<a href="#">CustomData</a>

The following is the XML Schema definition of the **CustomData.DataRows** element.

```
<xsd:element name="DataRows" type="DataRowsType" minOccurs="0" />
```

## 2.271.4 CustomData.DataSetName

Applies to [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomData.DataSetName** element specifies the name of a DataSet to use for a [CustomReportItem](#). This element is optional and MUST NOT be specified more than once. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). The value of this element MUST be one of the [DataSet.Name](#) attribute values of a [Report](#).

If the **CustomData** has an ancestor, the value of the **CustomData.DataSetName** element is interpreted as the **DataSet.Name** for the containing scope (**DataRegion**, [Group](#), or **Cell**).[<57>](#)

If the custom report item has a **DataRegion** element ancestor, the value of the **CustomData.DataSetName** element is ignored; the value of the **DataSetName** element ancestor of the **DataRegion** MUST be used instead. A **DataRegion** can be a [List](#), [Table](#), [Matrix](#), [Chart](#), [Tablix](#), [GaugePanel](#), or [Map](#).

If the custom report item has no **DataRegion** element ancestor and there is more than one dataset for the containing report, the value of the **CustomData.DataSetName** element MUST be specified.

If no value is specified for the **CustomData.DataSetName** element and the containing report has only one **DataSet** specified, the value of the **CustomData.DataSetName** element is interpreted as being the value of the **DataSet.Name** attribute for that dataset.

Following is the parent element of the **CustomData.DataSetName** element.

Parent elements
<a href="#">CustomData</a>

The following is the XML Schema definition of the **CustomData.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" />
```

## 2.271.5 CustomData.Filters

*Applies to [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomData.Filters** element specifies expressions that are applied as a **filter** to the data for each row of a [CustomReportItem](#). This element is optional and MUST NOT be specified more than once. If this element is present, it is of type [Filters](#).

Following is the parent element of the **CustomData.Filters** element.

Parent elements
<a href="#">CustomData</a>

The following is the XML Schema definition of the **CustomData.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

## 2.271.6 CustomData.SortExpressions

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomData.SortExpressions** element specifies a collection of expressions to be applied to the filtered data of a [CustomReportItem](#) to order the data. This element is optional and MUST NOT be specified more than once. If this element is present, it is of type [SortExpressions](#).

Following is the parent element of the **CustomData.SortExpressions** element.

Parent elements
<a href="#">CustomData</a>

The following is the XML Schema definition of the **CustomData.SortExpressions** element.

```
<xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
```

## 2.271.7 CustomData.DataColumnGroupings

*Applies to [RDL 2005/01](#)*

The **CustomData.DataColumnGroupings** element specifies the data column grouping hierarchy for a [CustomReportItem](#). This element MUST be specified exactly once and is of type [DataColumnGroupings](#).

The following is the parent element of the **CustomData.DataColumnGroupings** element.

Parent elements
<a href="#">CustomData</a>

The following is the XML Schema definition of the **CustomData.DataColumnGroupings** element.

```
<xsd:element name="DataColumnGroupings" type="DataColumnGroupingsType"
  minOccurs="0" />
```

## 2.271.8 CustomData.DataRowGroupings

*Applies to [RDL 2005/01](#)*

The **CustomData.DataRowGroupings** element specifies the data row grouping hierarchy for a [CustomReportItem](#). This element MUST be specified exactly once and is of type [DataRowGroupings](#).

Following is the parent element of the **CustomData.DataRowGroupings** element.

Parent elements
<a href="#">CustomData</a>

The following is the XML Schema definition of the **CustomData.DataRowGroupings** element.

```
<xsd:element name="DataRowGroupings" type="DataRowGroupingsType" minOccurs="0" />
```

## 2.271.9 CustomData.Relationship

*Applies to [RDL 2011/01](#)*

The **CustomData.Relationship** element specifies a **relationship** to use for correlating data in a [CustomData](#) with the data in the containing scope. The **CustomData.Relationship** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [Relationship](#). The **CustomData.Relationship** element is ignored if the dataset for this **CustomData** is the same as the data set for each containing scope. The **CustomData.Relationship** element MUST NOT be specified if there is no containing scope.

Following is the parent element of the **CustomData.Relationship** element.

Parent elements
CustomData

The following is the XML Schema definition of the **CustomData.Relationship** element.

```
<xsd:element name="Relationship" type="RelationshipType" minOccurs="0" />
```

## 2.272 DataColumnHierarchy

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **DataColumnHierarchy** parent element defines the hierarchy of [DataMember](#) elements for column data of a [CustomReportItem](#). The definition and use of column data is determined by the custom report item.

The following are the parent and child elements of the **DataColumnHierarchy** element.

Parent elements
<a href="#">CustomData</a>

Child element
<a href="#">DataColumnHierarchy.DataMembers</a>

The following is the XML Schema definition of the **DataColumnHierarchy** element in RDL 2008/01.

```
<xsd:complexType name="DataColumnHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataMembers" type="DataMembersType" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **DataColumnHierarchy** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="DataColumnHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataMembers" type="DataMembersType" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.272.1 DataColumnHierarchy.DataMembers

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **DataColumnHierarchy.DataMembers** element specifies members of the column data for a [CustomReportItem](#). This element MUST be specified exactly once and is of type [DataMembers](#).

Following is the parent element of the **DataColumnHierarchy.DataMembers** element.

Parent elements
<a href="#">DataColumnHierarchy</a>

The following is the XML Schema definition of the **DataColumnHierarchy.DataMembers** element.

```
<xsd:element name="DataMembers" type="DataMembersType"/>
```

## 2.273 DataMembers

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **DataMembers** element defines a collection of [DataMember](#) elements for a hierarchy.

The following are the parent and child elements of the **DataMembers** element.

Parent elements
<a href="#">DataColumnHierarchy</a>
<a href="#">DataRowHierarchy</a>

Child element
<a href="#">DataMembers.DataMember</a>

The following is the XML Schema definition of the **DataMembers** element in RDL 2008/01.

```
<xsd:complexType name="DataMembersType">
  <xsd:sequence>
    <xsd:element name="DataMember" type="DataMemberType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **DataMembers** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="DataMembersType">
  <xsd:sequence>
    <xsd:element name="DataMember" type="DataMemberType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.273.1 DataMembers.DataMember

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **DataMembers.DataMember** element specifies a [DataMember](#) element of a [DataMembers](#) element. The **DataMembers.DataMember** element **MUST** be specified at least once and is of type **DataMember**.

Following is the parent element of the **DataMembers.DataMember** element.

Parent elements
DataMembers

The following is the XML Schema definition of the **DataMembers.DataMember** element.

```
<xsd:element name="DataMember" type="DataMemberType" maxOccurs="unbounded" />
```

## 2.274 DataMember

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **DataMember** element defines the structure of a [CustomReportItem](#) hierarchy.

The following are the parent and child elements of the **DataMember** element.

Parent elements
<a href="#">DataMembers</a>

Child elements
<a href="#">DataMember.CustomProperties</a>
<a href="#">DataMember.DataMembers</a>
<a href="#">DataMember.Group</a>
<a href="#">DataMember.SortExpressions</a>
<a href="#">DataMember.Subtotal</a>

The following is the XML Schema definition of the **DataMember** element in RDL 2008/01.

```
<xsd:complexType name="DataMemberType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType"
      minOccurs="0" />
    <xsd:element name="Subtotal" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="DataMembers" type="DataMembersType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **DataMember** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="DataMemberType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType"
      minOccurs="0" />
    <xsd:element name="Subtotal" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="DataMembers" type="DataMembersType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```



</xsd:complexType>

### 2.274.1 DataMember.CustomProperties

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **DataMember.CustomProperties** element specifies custom information that is associated with a [DataMember](#) element. The **DataMember.CustomProperties** element is optional and MUST NOT be specified more than once. If this element is present, it is of type [CustomProperties](#).

Following is the parent element of the **DataMember.CustomProperties** element.

Parent elements
DataMember

The following is the XML Schema definition of the **DataMember.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

### 2.274.2 DataMember.DataMembers

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **DataMember.DataMembers** element specifies a collection of [DataMember](#) elements that form a hierarchy of data rows or columns for a [CustomReportItem](#). The **DataMember.DataMembers** element is optional and MUST NOT be specified more than once. If this element is present, it is of type [DataMembers](#).

Following is the parent element of the **DataMember.DataMembers** element.

Parent elements
DataMember

The following is the XML Schema definition of the **DataMember.DataMembers** element.

```
<xsd:element name="DataMembers" type="DataMembersType" minOccurs="0" />
```

### 2.274.3 DataMember.Group

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **DataMember.Group** element specifies the group for a [DataMember](#) element. The **DataMember.Group** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [Group](#). If a **DataMember** element has a **detail member** ancestor, the **DataMember.Group** element MUST NOT be specified. If this element is not present, the **DataMember** is a static member.

Following is the parent element of the **DataMember.Group** element.

Parent elements
DataMember

The following is the XML Schema definition of the **DataMember.Group** element.

```
<xsd:element name="Group" type="GroupType" minOccurs="0" />
```

## 2.274.4 DataMember.SortExpressions

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **DataMember.SortExpressions** element specifies a collection of expressions to order data of a [DataMember](#). The **DataMember.SortExpressions** element is optional and MUST NOT be specified more than once. This element MUST NOT be specified for static members. If this element is present, it is of type [SortExpressions](#).

Following is the parent element of the **DataMember.SortExpressions** element.

Parent elements
<a href="#">DataMember</a>

The following is the XML Schema definition of the **DataMember.SortExpressions** element.

```
<xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
```

## 2.274.5 DataMember.Subtotal

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **DataMember.Subtotal** element is ignored.

Following is the parent element of the **DataMember.Subtotal** element.

Parent elements
<a href="#">DataMember</a>

The following is the XML Schema definition of the **DataMember.Subtotal** element.

```
<xsd:element name="Subtotal" type="xsd:boolean" minOccurs="0" />
```

## 2.275 DataRowHierarchy

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **DataRowHierarchy** parent element defines the hierarchy of [DataMember](#) elements for row data of a [CustomReportItem](#). The definition and use of row data is determined by the custom report item.

The following are the parent and child elements of the **DataRowHierarchy** element.

Parent elements
<a href="#">CustomData</a>

<b>Child element</b>
----------------------

<a href="#">DataRowHierarchy.DataMembers</a>
--

The following is the XML Schema definition of the **DataRowHierarchy** element in RDL 2008/01.

```
<xsd:complexType name="DataRowHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataMembers" type="DataMembersType" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **DataRowHierarchy** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="DataRowHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataMembers" type="DataMembersType" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.275.1 DataRowHierarchy.DataMembers

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **DataRowHierarchy.DataMembers** element specifies the members of [CustomReportItem](#) rows. This element MUST be specified exactly once and is of type [DataMembers](#).

Following is the parent element of the **DataRowHierarchy.DataMembers** element.

<b>Parent elements</b>
------------------------

<a href="#">DataRowHierarchy</a>
----------------------------------

The following is the XML Schema definition of the **DataRowHierarchy.DataMembers** element.

```
<xsd:element name="DataMembers" type="DataMembersType"/>
```

### 2.276 DataRows

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **DataRows** parent element defines the rows of data for a [CustomReportItem](#) as a collection of [DataRow](#) elements.

The following are the parent and child elements of the **DataRows** element.

<b>Parent elements</b>
------------------------

<a href="#">CustomData</a>
----------------------------

<b>Child element</b>
----------------------

<a href="#">DataRows.DataRow</a>
----------------------------------

The following is the XML Schema definition of the **DataRows** element in RDL 2005/01 and RDL 2008/01.

```
<xsd:complexType name="DataRowsType">
  <xsd:sequence>
    <xsd:element name="DataRow" type="DataRowType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **DataRows** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="DataRowsType">
  <xsd:sequence>
    <xsd:element name="DataRow" type="DataRowType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.276.1 DataRows.DataRow

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **DataRows.DataRow** element specifies a row in the body of a [CustomReportItem](#). This element MUST be specified at least once and is of type [DataRow](#).

The quantity of **DataRow** elements MUST equal the quantity of [DataMember](#) elements in the [DataRowHierarchy](#) that do not have a **DataMember** element descendant.

Following is the parent element of the **DataRows.DataRow** element.

<b>Parent elements</b>
------------------------

<a href="#">DataRows</a>
--------------------------

The following is the XML Schema definition of the **DataRows.DataRow** element.

```
<xsd:element name="DataRow" type="DataRowType" maxOccurs="unbounded" />
```

## 2.277 DataRow

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **DataRow** parent element defines a collection of [DataCell](#) elements that contain data within each row for a [CustomReportItem](#).

The following are the parent and child elements of the **DataRow** element.

<b>Parent elements</b>
------------------------

<a href="#">DataRows</a>
--------------------------

<b>Child element</b>
----------------------

<a href="#">DataRow.DataCell</a>
----------------------------------

The following is the XML Schema definition of the **DataRow** element in RDL 2005/01 and RDL 2008/01.

```
<xsd:complexType name="DataRowType">
  <xsd:sequence>
    <xsd:element name="DataCell" type="DataCellType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **DataRow** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="DataRowType">
  <xsd:sequence>
    <xsd:element name="DataCell" type="DataCellType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.277.1 DataRow.DataCell

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **DataRow.DataCell** element specifies a cell of data values in a [CustomReportItem](#). This element MUST be specified at least once and MUST be of type [DataCell](#). The quantity of **DataCell** elements in each [DataRow](#) MUST equal the quantity of [DataMember](#) elements in the [DataColumnHierarchy](#) that do not have a **DataMember** element descendant.

Following is the parent element of the **DataRow.DataCell** element.

<b>Parent elements</b>
------------------------

<a href="#">DataRow</a>
-------------------------

The following is the XML Schema definition of the **DataRow.DataCell** element.

```
<xsd:element name="DataCell" type="DataCellType" maxOccurs="unbounded" />
```

### 2.278 DataCell

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **DataCell** parent element defines a collection of [DataValue](#) elements for a cell in a [CustomReportItem](#). The purpose and use of the content of a data cell are determined by the custom

report item code. Every **DataCell** relates to a combination of leaf-node [DataMembers](#) in both [DataRowHierarchy](#) and [DataColumnHierarchy](#).

The following are the parent and child elements of the **DataCell** element.

Parent elements
<a href="#">DataRow</a>

Child elements
<a href="#">DataCell.DataValue</a>

**Applies to** [RDL 2011/01](#)

Child elements
<a href="#">DataCell.DataSetName</a>
<a href="#">DataCell.Relationships</a>

The following is the XML Schema definition of the **DataCell** parent element in RDL 2005/01 and RDL 2008/01.

```
<xsd:complexType name="DataCellType">
  <xsd:sequence>
    <xsd:element name="DataValue" type="DataValueType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **DataCell** parent element in RDL 2010/01 and RDL 2016/01.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:complexType name="DataCellType">
  <xsd:sequence>
    <xsd:element name="DataValue" type="DataValueType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.278.1 DataCell.DataValue

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **DataCell.DataValue** element specifies a data item for a cell of a [CustomReportItem](#). This element MUST be specified at least once. This element is of type [DataValue](#).

Following is the parent element of the **DataCell.DataValue** element.

<b>Parent elements</b>
<a href="#">DataCell</a>

The following is the XML Schema definition of the **DataCell.DataValue** element.

```
<xsd:element name="DataValue" type="DataValueType" maxOccurs="unbounded" />
```

## 2.278.2 DataCell.DataSetName

**Applies to** [RDL 2011/01](#)

The **DataCell.DataSetName** element specifies which [DataSet](#) to use for a [DataCell](#). The **DataCell.DataSetName** element is optional and MUST NOT be specified more than once.

If the **DataCell.DataSetName** element is specified, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is a **CLS-compliant identifier** [UTR15]. If this element is not present, its value is interpreted as the **DataSetName** of the containing scope. The **DataCell.DataSetName** element MUST be specified unless all containing scopes have the same **DataSet**. This element is ignored if the **DataSet** for this **DataCell** is the same as the **DataSet** for all containing scopes. [<58><59>](#)

Following is the parent element of the **DataCell.DataSetName** element.

<b>Parent elements</b>
DataCell

The following is the XML Schema definition of the **DataCell.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0"/>
```

## 2.278.3 DataCell.Relationships

**Applies to** [RDL 2011/01](#)

The **DataCell.Relationships** element specifies a collection of the **relationships** to use for correlating data in this [DataCell](#) with the data in the containing scopes. The **DataCell.Relationships** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [Relationships](#).

This element is ignored if the [DataSet](#) for this **DataCell** is the same as the **DataSet** for each containing scope.

Following is the parent element of the **DataCell.Relationships** element.

<b>Parent elements</b>
DataCell

The following is the XML Schema definition of the **DataCell.Relationships** element.

```
<xsd:element name="Relationships" type="RelationshipsType" minOccurs="0" />
```

## 2.279 DataValue

The **DataValue** element defines a single value within a cell of a [CustomReportItem](#) or, in [RDL 2003/10](#) and [RDL 2005/01](#), a data value instance of a data point of a [Chart](#).

The following are the parent and child elements of the **DataValue** element.

Parent elements
<a href="#">DataCell</a>
<a href="#">DataValues</a>

Child elements
<a href="#">DataValue.Name</a>
<a href="#">DataValue.Value</a>

The following is the XML Schema definition of the **DataValue** element.

```
<xsd:complexType name="DataValueType">>
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.279.1 DataValue.Name

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **DataValue.Name** element specifies a name to identify a datum value in a cell of a [CustomReportItem](#) or a name to identify a data value instance of a data point of a [Chart](#). The **DataValue.Name** element is optional and MUST NOT be specified more than once. If this element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **DataValue.Name** element.

Parent elements
<a href="#">DataValue</a>

The following is the XML Schema definition of the **DataValue.Name** element.

```
<xsd:element name="Name" type="xsd:string" minOccurs="0" />
```

### 2.279.2 DataValue.Value

The **DataValue.Value** element specifies a datum value in a cell of a [CustomReportItem](#) or a data value instance of a data point of a [Chart](#). This element MUST be specified exactly once and MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **Variant**.



Following is the parent element of the **DataValue.Value** element.

Parent elements
<a href="#">DataValue</a>

The following is the XML Schema definition of the **DataValue.Value** element.

```
<xsd:element name="Value" type="xsd:string" />
```

## 2.280 DataColumnGroupings

**Applies to** [RDL 2005/01](#)

The **DataColumnGroupings** parent element defines a collection of [DataGroupings](#) for column data of a [CustomReportItem](#).

The following are the parent and child elements of the **DataColumnGroupings** element.

Parent elements
<a href="#">CustomData</a>

Child element
<a href="#">DataColumnGroupings.DataGroupings</a>

The following is the XML Schema definition of the **DataColumnGroupings** element.

```
<xsd:complexType name="DataColumnGroupingsType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataGroupings" type="DataGroupingsType" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.280.1 DataColumnGroupings.DataGroupings

**Applies to** [RDL 2005/01](#)

The **DataColumnGroupings.DataGroupings** element specifies data groupings for a column of data in a [CustomReportItem](#). This element MUST be specified exactly once and is of type [DataGroupings](#).

Following is the parent element of the **DataColumnGroupings.DataGroupings** element.

Parent elements
<a href="#">DataColumnGroupings</a>

The following is the XML Schema definition of the **DataColumnGroupings.DataGroupings** element.

```
<xsd:element name="DataGroupings" type="DataGroupingsType" />
```

## 2.281 DataGroupings

**Applies to** [RDL 2005/01](#)

The **DataGroupings** parent element defines a collection of [DataGrouping](#) elements.

The following are the parent and child elements of the **DataGroupings** element.

Parent elements
<a href="#">DataColumnGroupings</a>
<a href="#">DataRowGroupings</a>

Child element
<a href="#">DataGroupings.DataGrouping</a>

The following is the XML Schema definition of the **DataGroupings** element.

```
<xsd:complexType name="DataGroupingsType">
  <xsd:sequence>
    <xsd:element name="DataGrouping" type="DataGroupingType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.281.1 DataGroupings.DataGrouping

**Applies to** [RDL 2005/01](#)

The **DataGroupings.DataGrouping** element specifies a grouping of data. This element **MUST** be specified at least once and is of type [DataGrouping](#). If more than one **DataGrouping** element is specified, the value of all child [DataGrouping.Static](#) elements **MUST** be true.

Following is the parent element of the **DataGroupings.DataGrouping** element.

Parent elements
<a href="#">DataGroupings</a>

The following is the XML Schema definition of the **DataGroupings.DataGrouping** element.

```
<xsd:element name="DataGrouping" type="DataGroupingType" maxOccurs="unbounded" />
```

## 2.282 DataGrouping

**Applies to** [RDL 2005/01](#)

The **DataGrouping** parent element defines the structure of a [CustomReportItem](#) data hierarchy.

The following are the parent and child elements of the **DataGrouping** element.

<b>Parent elements</b>
<a href="#">DataGroupings</a>

<b>Child elements</b>
<a href="#">DataGrouping.CustomProperties</a>
<a href="#">DataGrouping.DataGroupings</a>
<a href="#">DataGrouping.Grouping</a>
<a href="#">DataGrouping.Sorting</a>
<a href="#">DataGrouping.Static</a>
<a href="#">DataGrouping.Subtotal</a>

The following is the XML Schema definition of the **DataGrouping** element.

```
<xsd:complexType name="DataGroupingType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Static" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Grouping" type="GroupingType" minOccurs="0" />
    <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
    <xsd:element name="Subtotal" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="DataGroupings" type="DataGroupingsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.282.1 DataGrouping.CustomProperties

**Applies to** [RDL 2005/01](#)

The **DataGrouping.CustomProperties** element specifies custom information that is associated with a [DataGrouping](#) element. The **DataGrouping.CustomProperties** element is optional and MUST NOT be specified more than once. If this element is present, it is of type [CustomProperties](#).

Following is the parent element of the **DataGrouping.CustomProperties** element.

<b>Parent elements</b>
<a href="#">DataGrouping</a>

The following is the XML Schema definition of the **DataGrouping.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

### 2.282.2 DataGrouping.DataGroupings

**Applies to** [RDL 2005/01](#)

The **DataGrouping.DataGroupings** element specifies a collection of [DataGrouping](#) elements that form a hierarchy of rows or columns for a [CustomReportItem](#). The **DataGrouping.DataGroupings** element is optional and MUST NOT be specified more than once. If this element is present, it is of type [DataGroupings](#).

Following is the parent element of the **DataGrouping.DataGroupings** element.

Parent elements
DataGrouping

The following is the XML Schema definition of the **DataGrouping.DataGroupings** element.

```
<xsd:element name="DataGroupings" type="DataGroupingsType" minOccurs="0" />
```

### 2.282.3 DataGrouping.Grouping

*Applies to [RDL 2005/01](#)*

The **DataGrouping.Grouping** element specifies the group for a [DataGrouping](#) element. The **DataGrouping.Grouping** element is optional and MUST NOT be specified more than once. If this element is present, it is of type [Grouping](#).

If the **DataGrouping.Grouping** element is present, the value of the [DataGrouping.Static](#) element MUST be false. If the **DataGrouping.Grouping** element is not present, the value of the **DataGrouping.Static** element MUST be true.

Following is the parent element of the **DataGrouping.Grouping** element.

Parent elements
DataGrouping

The following is the XML Schema definition of the **DataGrouping.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" minOccurs="0" />
```

### 2.282.4 DataGrouping.Sorting

*Applies to [RDL 2005/01](#)*

The **DataGrouping.Sorting** element specifies information to order data in a [DataGrouping](#) element. The **DataGrouping.Sorting** element is optional and MUST NOT be specified more than once. If the **DataGrouping.Sorting** element is specified, it is of type [Sorting](#). If the **DataGrouping.Sorting** element is specified, the value of the [DataGrouping.Static](#) element MUST be false.

Following is the parent element of the **DataGrouping.Sorting** element.

Parent elements
DataGrouping

The following is the XML Schema definition of the **DataGrouping.Sorting** element.

```
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
```

## 2.282.5 DataGrouping.Static

**Applies to** [RDL 2005/01](#)

The **DataGrouping.Static** element specifies whether a [DataGrouping](#) element is fixed in the RDL or is dynamically generated through a **group expression**. The **DataGrouping.Static** element is optional and MUST NOT be specified more than once.

If the **DataGrouping.Static** element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2](#) section 3.2.2). If the value of this element is true, the **DataGrouping** is static and not dynamic. If the **DataGrouping.Static** element is not present, its value is interpreted as false.

If the value of the **DataGrouping.Static** element is false, a [DataGrouping.Grouping](#) element MUST be specified. If the value of the **DataGrouping.Static** element is true, a **DataGrouping.Grouping** element MUST NOT be specified.

If the value of the **DataGrouping.Static** element is true, the **DataGrouping** element MUST NOT have **DataGrouping** element descendants. If the value of a particular **DataGrouping.Static** element is true, all peer **DataGrouping.Static** values in the same parent [DataGroupings](#) collection MUST also be true.

Following is the parent element of the **DataGrouping.Static** element.

Parent elements
<a href="#">DataGrouping</a>

The following is the XML Schema definition of the **DataGrouping.Static** element.

```
<xsd:element name="Static" type="xsd:boolean" minOccurs="0" />
```

## 2.282.6 DataGrouping.Subtotal

**Applies to** [RDL 2005/01](#)

The **DataGrouping.Subtotal** element is ignored.

Following is the parent element of the **DataGrouping.Subtotal** element.

Parent elements
<a href="#">DataGrouping</a>

The following is the XML Schema definition of the **DataGrouping.Subtotal** element.

```
<xsd:element name="Subtotal" type="xsd:boolean" minOccurs="0" />
```

## 2.283 DataRowGroupings

**Applies to** [RDL 2005/01](#)

The **DataRowGroupings** parent element defines a collection of [DataGroupings](#) for the row data of a [CustomReportItem](#).

The following are the parent and child elements of the **DataRowGroupings** element.

Parent elements
<a href="#">CustomData</a>

Child element
<a href="#">DataRowGroupings.DataGroupings</a>

The following is the XML Schema definition of the **DataRowGroupings** element.

```
<xsd:complexType name="DataRowGroupingsType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataGroupings" type="DataGroupingsType" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.283.1 DataRowGroupings.DataGroupings

**Applies to** [RDL 2005/01](#)

The **DataRowGroupings.DataGroupings** element specifies data groupings for a row of data in a [CustomReportItem](#). This element MUST be specified exactly once and is of type [DataGroupings](#).

Following is the parent element of the **DataRowGroupings.DataGroupings** element.

Parent elements
<a href="#">DataRowGroupings</a>

The following is the XML Schema definition of the **DataRowGroupings.DataGroupings** element.

```
<xsd:element name="DataGroupings" type="DataGroupingsType" />
```

### 2.284 CustomReportItem

The **CustomReportItem** element defines a **report item** that is not natively detailed in RDL and that requires additional code to render.

Information that is in addition to information that is specified by using the [DataSet](#) in the [CustomData](#) element of the **CustomReportItem** in [RDL 2005/01](#), or information that is in addition to information that is specified by using the [CustomData.DataSetName](#) element of the **CustomReportItem** in [RDL 2008/01](#), [RDL 2010/01](#), or [RDL 2016/01](#), can be specified by using [CustomProperty](#) elements that are either children of the **CustomReportItem** element or children of the [DataMember](#) elements that are descendants of the **CustomReportItem**.

In RDL 2005/01, RDL 2008/01, RDL 2010/01, and RDL 2016/01, a **custom report item** MUST NOT appear in the page header or page footer of a [Page](#).

Renderers that do not support the specific custom report item render one of the following:

- In [RDL 2003/10](#), the report items that are specified by the [CustomReportItem.ReportItems](#) element.
- In RDL 2005/01, RDL 2008/01, RDL 2010/01, and RDL 2016/01, the report item that is specified by the [CustomReportItem.AltReportItem](#) element.

A report item that is specified as a child of the **CustomReportItem.AltReportItem** element is available as a referenced report item for RDL file validation purposes. The **CustomReportItem.Name** attribute MUST NOT be the same as the **Name** attribute of a report item that is specified as a child of an **AltReportItem** element.

The following are the parent elements of the **CustomReportItem** element.

Parent elements
<a href="#">ReportItems</a>

The following are additional parent elements of the **CustomReportItem** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Parent elements
<a href="#">CellContents</a>

The following are the attributes of the **CustomReportItem** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Attributes
<a href="#">CustomReportItem.Name</a>

The following are the child elements of the **CustomReportItem** element.

Child elements
<a href="#">CustomReportItem.Type</a>

The following are additional child elements of the **CustomReportItem** element in RDL 2003/10.

Child elements
CustomReportItem.ReportItems

The following are additional child elements of the **CustomReportItem** element in RDL 2005/01.

Child elements
<a href="#">CustomReportItem.Label</a>

The following are additional child elements of the **CustomReportItem** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Child elements
<a href="#">CustomReportItem.ActionInfo</a>

Child elements
CustomReportItem.AltReportItem
<a href="#">CustomReportItem.Bookmark</a>
<a href="#">CustomReportItem.CustomData</a>
<a href="#">CustomReportItem.CustomProperties</a>
<a href="#">CustomReportItem.DataElementName</a>
<a href="#">CustomReportItem.DataElementOutput</a>
<a href="#">CustomReportItem.DocumentMapLabel</a>
<a href="#">CustomReportItem.Height</a>
<a href="#">CustomReportItem.Left</a>
<a href="#">CustomReportItem.RepeatWith</a>
<a href="#">CustomReportItem.Style</a>
<a href="#">CustomReportItem.ToolTip</a>
<a href="#">CustomReportItem.Top</a>
CustomReportItem.Type
<a href="#">CustomReportItem.Visibility</a>
<a href="#">CustomReportItem.Width</a>
<a href="#">CustomReportItem.ZIndex</a>

The following is the XML Schema definition of the **CustomReportItem** element in RDL 2003/10.

```

<xsd:complexType name="CustomReportItemType">
  <xsd:all>
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
          <xsd:enumeration value="Auto"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:attribute name="Type" type="xsd:normalizedString" use="required" />
</xsd:complexType>

```



The following is the XML Schema definition of the **CustomReportItem** element in RDL 2005/01.

```
<xsd:complexType name="CustomReportItemType" >
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Type" type="xsd:string" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0"/>
    <xsd:element name="AltReportItem" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="CustomData" type="CustomDataType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
          <xsd:enumeration value="Auto"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
  <xsd:any namespace="##other" processContents="skip"/>
</xsd:complexType>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
```

The following is the XML Schema definition of the **CustomReportItem** element in RDL 2008/01.

```
<xsd:complexType name="CustomReportItemType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Type" type="xsd:string" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="AltReportItem" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="CustomData" type="CustomDataType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

```

    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **CustomReportItem** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="CustomReportItemType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Type" type="xsd:string" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="AltReportItem" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="CustomData" type="CustomDataType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.284.1 CustomReportItem.Name

The **CustomReportItem.Name** attribute specifies a unique identifier for a [CustomReportItem](#). The **CustomReportItem.Name** attribute MUST be specified, and its value MUST be a [NormalizedString](#) that is a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique among all report item names and scope names.

Following is the parent element of the **CustomReportItem.Name** attribute.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

## 2.284.2 CustomReportItem.Type

In [RDL 2003/10](#), **CustomReportItem.Type** is an attribute that specifies an identifier for a [CustomReportItem](#), the value of which is interpreted by a report design tool or server. The **CustomReportItem.Type** attribute MUST be specified and its value MUST be a [NormalizedString](#).

In [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#), **CustomReportItem.Type** is an element that specifies a reference name that enables the identification of the custom program that renders a CustomReportItem. The **CustomReportItem.Type** element MUST be specified and is of type [String](#) ([XMLSCHEMA2/2] section 3.2.1).

Following is the parent element of the **CustomReportItem.Type** attribute in RDL 2003/10, and the **CustomReportItem.Type** element in RDL 2005/01, RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Parent elements
<a href="#">CustomReportItem</a>

The following is the XML Schema definition of the **CustomReportItem.Type** attribute in RDL 2003/10.

```
<xsd:attribute name="Type" type="xsd:normalizedString" use="required" />
```

The following is the XML Schema definition of the **CustomReportItem.Type** element in RDL 2005/01, RDL 2008/01, RDL 2010/01, and RDL 2016/01.

```
<xsd:element name="Type" type="xsd:string" />
```

## 2.284.3 CustomReportItem.ActionInfo

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomReportItem.ActionInfo** element is ignored.

Following is the parent element of the **CustomReportItem.ActionInfo** element.

Parent elements
<a href="#">CustomReportItem</a>

The following is the XML Schema definition of the **CustomReportItem.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

## 2.284.4 CustomReportItem.AltReportItem

*Applies to [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomReportItem.AltReportItem** element specifies a report item element that is rendered if a [CustomReportItem](#) is not supported by the renderer. The **CustomReportItem.AltReportItem** element is optional and MUST NOT be specified more than once.

If this element is specified, it is of type [ReportItems](#), and it MUST have a single ReportItem element child specified that is not of type **CustomReportItem**.

If the **CustomReportItem.AltReportItem** element is not specified, the value of the **ReportItem** element child is interpreted as a **Rectangle** element with no border and with a transparent background. [<60>](#)

Any restrictions on the placement or use of a report item apply when the report item is specified as a descendant of the **CustomReportItem.AltReportItem** element.

Following is the parent element of the **CustomReportItem.AltReportItem** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.AltReportItem** element.

```
<xsd:element name="AltReportItem" type="ReportItemsType" minOccurs="0" />
```

## 2.284.5 CustomReportItem.Bookmark

*Applies to [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomReportItem.Bookmark** element specifies the bookmark of a [CustomReportItem](#). This element is optional and MUST NOT be specified more than once. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **CustomReportItem.Bookmark** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.Bookmark** element.

```
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
```

## 2.284.6 CustomReportItem.CustomData

*Applies to [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **CustomReportItem.CustomData** element specifies the data and data manipulation for a [CustomReportItem](#). This element is optional and MUST NOT be specified more than once. If this element is present, it is of type [CustomData](#).

Following is the parent element of the **CustomReportItem.CustomData** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.CustomData** element.

```
<xsd:element name="CustomData" type="CustomDataType" minOccurs="0" />
```

## 2.284.7 CustomReportItem.CustomProperties

Applies to [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomReportItem.CustomProperties** element specifies extended information in a collection of name/value pairs. This element is optional and MUST NOT be specified more than once. If this element is present, it is of type [CustomProperties](#).

Following is the parent element of the **CustomReportItem.CustomProperties** element.

Parent elements
<a href="#">CustomReportItem</a>

The following is the XML Schema definition of the **CustomReportItem.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

## 2.284.8 CustomReportItem.DataElementName

The **CustomReportItem.DataElementName** is an optional element that SHOULD be ignored and that MUST NOT be specified more than once. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is a **CLS-compliant identifier** [\[UTR15\]](#) that is unique within the same data element hierarchy.

Following is the parent element of the **CustomReportItem.DataElementName** element.

Parent elements
<a href="#">CustomReportItem</a>

The following is the XML Schema definition of the **CustomReportItem.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

## 2.284.9 CustomReportItem.DataElementOutput

The **CustomReportItem.DataElementOutput** is an optional element that SHOULD be ignored and that MUST NOT be specified more than once. If this element is specified, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

- **Output**
- **NoOutput**
- **Auto**
- **ContentsOnly**

Following is the parent element of the **CustomReportItem.DataElementOutput** element.

Parent elements
<a href="#">CustomReportItem</a>

The following is the XML Schema definition of the **CustomReportItem.DataElementOutput** element.

```
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.284.10 CustomReportItem.DocumentMapLabel

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomReportItem.DocumentMapLabel** element specifies a document map label to identify a [CustomReportItem](#) within a rendered report. This element is optional and MUST NOT be specified more than once. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **Variant**.

Following is the parent element of the **CustomReportItem.DocumentMapLabel** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.DocumentMapLabel** element.

```
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
```

## 2.284.11 CustomReportItem.Height

The **CustomReportItem.Height** element specifies the height of a [CustomReportItem](#). This element is optional and MUST NOT be specified more than once. If this element is present, it is of type [RdlSize](#).

If the **CustomReportItem.Height** element is not present, if the custom report item is not supported by the renderer, and if the [CustomReportItem.AltReportItem](#) element specifies a [Tablix](#), then the height of the **CustomReportItem** is interpreted to be the maximum of the height of the tablix and the value of the **CustomReportItem.Height** element.

Under any other circumstances, if the **CustomReportItem.Height** element is not present, its value is interpreted as the **Height** element child of the immediate ancestor report item element minus the value of the [CustomReportItem.Top](#) element.

**Note** The height of a **CustomReportItem** is fixed and cannot be adjusted by the report item code.

Following is the parent element of the **CustomReportItem.Height** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.Height** element.

```
<xsd:element name="Height" type="SizeType" minOccurs="0" />
```

## 2.284.12 CustomReportItem.Label

**Applies to** [RDL 2005/01](#)

The **CustomReportItem.Label** element specifies a document map label to identify a [CustomReportItem](#) within a rendered report. This element is optional and MUST NOT be specified more than once. If this element is specified, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **VARIANT**.

Following is the parent element of the **CustomReportItem.Label** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
```

## 2.284.13 CustomReportItem.Left

The **CustomReportItem.Left** element specifies the distance of a [CustomReportItem](#) from the left of a containing object. The **CustomReportItem.Left** element is optional and MUST NOT be specified more than once. If this element is present, it is of type [RdlSize](#). If this element is not present, its value is interpreted as 0.

Following is the parent element of the **CustomReportItem.Left** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.Left** element.

```
<xsd:element name="Left" type="SizeType" minOccurs="0" />
```

## 2.284.14 CustomReportItem.RepeatWith

The **CustomReportItem.RepeatWith** element specifies a data region that the [CustomReportItem](#) is repeated on for every [Page](#) upon which the data region is rendered. The **CustomReportItem.RepeatWith** element MUST be the name of a data region within the same [ReportItems](#) element ancestor as the **CustomReportItem**.

The **CustomReportItem.RepeatWith** element is optional and MUST NOT be specified more than once. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1). If the **CustomReportItem** element has an element ancestor of type [PageSection](#) or an element ancestor of type [PageHeaderFooter](#), the **CustomReportItem.RepeatWith** element is ignored.

Following is the parent element of the **CustomReportItem.RepeatWith** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.RepeatWith** element.

```
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
```

## 2.284.15 CustomReportItem.ReportItems

**Applies to** [RDL 2003/10](#)

The **CustomReportItem.ReportItems** element specifies a collection of report item elements that are rendered if the [CustomReportItem](#) is not supported by the renderer.

The **CustomReportItem.ReportItems** element is optional and MUST NOT be specified more than once. If this element is present, it is of type [ReportItems](#). If the **CustomReportItem.ReportItems** element is not present, the **ReportItems** element child is interpreted as a [Rectangle](#) element with no border and with a transparent background.

The **Top**, **Left**, **Height**, and **Width** elements of the **CustomReportItem** element are used only if **ReportItem** elements in **CustomReportItem.ReportItems** are not being rendered.

Following is the parent element of the **CustomReportItem.ReportItems** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.ReportItems** element.

```
<xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
```

## 2.284.16 CustomReportItem.Style

The **CustomReportItem.Style** element specifies the [Style](#) element of a [CustomReportItem](#) element. The **CustomReportItem.Style** element is optional and MUST NOT be specified more than once. If this element is present, it is of type **Style**.

Following is the parent element of the **CustomReportItem.Style** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

## 2.284.17 CustomReportItem.ToolTip

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)



The **CustomReportItem.ToolTip** element specifies a tooltip for a [CustomReportItem](#) element. The element can also be used to render alternative text (alt text) that is specified as an **alt** attribute in an HTML report. The **CustomReportItem.ToolTip** element is optional. This MUST NOT be specified more than once. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **Variant**.

Following is the parent element of the **CustomReportItem.ToolTip** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

### 2.284.18 CustomReportItem.Top

The **CustomReportItem.Top** element specifies the distance of a [CustomReportItem](#) from the top of a containing object. This element is optional and MUST NOT be specified more than once. If this element is present, it is of type [RdlSize](#). If the **CustomReportItem.Top** element is not present, its value is interpreted as 0.

Following is the parent element of the **CustomReportItem.Top** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.Top** element.

```
<xsd:element name="Top" type="SizeType" minOccurs="0" />
```

### 2.284.19 CustomReportItem.Visibility

The **CustomReportItem.Visibility** element specifies the presentational presence of a [CustomReportItem](#) element. The **CustomReportItem.Visibility** element is optional and MUST NOT be specified more than once. If this element is present, it is of type [Visibility](#).

Following is the parent element of the **CustomReportItem.Visibility** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

## 2.284.20 CustomReportItem.Width

The **CustomReportItem.Width** element specifies the width of a [CustomReportItem](#) element. The **CustomReportItem.Width** element is optional and MUST NOT be specified more than once. If this element is present, it is of type [RdlSize](#).

If the **CustomReportItem.Width** element is not present, if the custom report item is not supported by the renderer, and if the [CustomReportItem.AltReportItem](#) element specifies a tablix, the width of the custom report item is interpreted to be the maximum of the width of the tablix and the value of the **CustomReportItem.Width** element.

Under any other circumstances, if the **CustomReportItem.Width** element is not present, its value is interpreted as the **Width** element child of the immediate ancestor report item element minus the value of the [CustomReportItem.Left](#) element.

**Note** The width of a custom report item is fixed and cannot be adjusted by the report item code.

Following is the parent element of the **CustomReportItem.Width** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.Width** element.

```
<xsd:element name="Width" type="SizeType" minOccurs="0" />
```

## 2.284.21 CustomReportItem.ZIndex

The **CustomReportItem.ZIndex** element specifies the **ZIndex** of a [CustomReportItem](#). This element is optional and MUST NOT be specified more than once. If this element is specified, its value MUST be an **UnsignedInt**.

Following is the parent element of the **CustomReportItem.ZIndex** element.

Parent elements
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.ZIndex** element.

```
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
```

## 2.285 ActionInfo

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ActionInfo** element specifies a list of **actions** that are for a report item.

In [RDL 2003/10](#) and [RDL 2005/01](#), the [Action](#) element specifies a list of actions for a report item.

The following are the parent and child elements of the **ActionInfo** element.

Parent elements
<a href="#">Image</a>

<b>Parent elements</b>
<a href="#">Line</a>
<a href="#">Rectangle</a>
<a href="#">Subreport</a>
<a href="#">Textbox</a>
<a href="#">TextRun</a>
<a href="#">Tablix</a>
<a href="#">Chart</a>
<a href="#">ChartStripLine</a>
<a href="#">ChartDataLabel</a>
<a href="#">ChartDataPoint</a>
<a href="#">ChartEmptyPoints</a>
<a href="#">ChartItemInLegend</a>
<a href="#">ChartLegendColumn</a>
<a href="#">ChartTitle</a>
<a href="#">ChartLegendCustomItem</a>
<a href="#">ChartLegendCustomItemCell</a>
<a href="#">GaugePanel</a>
<a href="#">GaugeImage</a>
<a href="#">GaugeLabel</a>
<a href="#">LinearGauge</a>
<a href="#">LinearScale</a>
<a href="#">LinearPointer</a>
<a href="#">ScaleRange</a>
<a href="#">NumericIndicator</a>
<a href="#">RadialGauge</a>
<a href="#">RadialScale</a>
<a href="#">RadialPointer</a>
<a href="#">StateIndicator</a>
<a href="#">Map</a>
<a href="#">MapColorScale</a>
<a href="#">MapDistanceScale</a>
<a href="#">MapLineTemplate</a>

Parent elements
<a href="#">MapMarkerTemplate</a>
<a href="#">MapPolygonTemplate</a>
<a href="#">MapLegend</a>
<a href="#">MapTitle</a>
<a href="#">CustomReportItem</a>

Child elements
<a href="#">ActionInfo.Actions</a>

The following is the XML Schema definition of the **ActionInfo** element in RDL 2008/01.

```
<xsd:complexType name="ActionInfoType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Actions" type="ActionsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **ActionInfo** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ActionInfoType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Actions" type="ActionsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.285.1 ActionInfo.Actions

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ActionInfo.Actions** element specifies actions that are for a report item. This element is optional and is of type [Actions](#).

Following is the parent element of the **ActionInfo.Actions** element.

Parent elements
<a href="#">ActionInfo</a>

The following is the XML Schema definition of the **ActionInfo.Actions** element.

```
<xsd:element name="Actions" type="ActionsType" minOccurs="0">
```

## 2.286 Actions

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Actions** element specifies the [Action](#) elements for a report item. The **Actions** element is optional.

The following are the parent and child elements of the **Actions** element.

Parent elements
<a href="#">ActionInfo</a>

Child elements
<a href="#">Actions.Action</a>

The following is the XML Schema definition of the **Actions** element in RDL 2008/01.

```
<xsd:complexType name="ActionTypes">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Action" type="ActionType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **Actions** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ActionTypes">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Action" type="ActionType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.286.1 Actions.Action

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Actions.Action** element specifies an action that is associated with a report item. This element **MUST** be specified at least once within an [Actions](#) collection. This element is of type [Action](#).

Following is the parent element of the **Actions.Action** element.

Parent elements
Actions

The following is the XML Schema definition of the **Actions.Action** element.

```
<xsd:element name="Action" type="ActionType" minOccurs="1" maxOccurs="unbounded" />
```

## 2.287 Action

The **Action** element specifies a hyperlink, bookmark link, or drillthrough **action** that is associated with a report item. This element MUST be specified at least once within an [Actions](#) collection. The **Action** element MUST contain one and only one of the following:

- [Action.Hyperlink](#)
- [Action.BookmarkLink](#)
- [Action.Drillthrough](#)

The following are the parent elements of the **Action** element.

Parent elements
<a href="#">Image</a>
<a href="#">Line</a>
<a href="#">Rectangle</a>
<a href="#">Subreport</a>
<a href="#">Textbox</a>
<a href="#">Chart</a>

The following are additional parent elements of the **Action** element in [RDL 2003/10](#) and [RDL 2005/01](#).

Parent elements
<a href="#">List</a>
<a href="#">Matrix</a>
<a href="#">Table</a>
<a href="#">DataPoint</a>

The following are additional parent elements of the **Action** element in [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#).

Parent elements
Actions

The following are the child elements of the **Action** element.

Child elements
Action.BookmarkLink
Action.Drillthrough
Action.Hyperlink

The following are additional child elements of the **Action** element in RDL 2005/01.

## Child elements

[Action.Label](#)

The following is the XML Schema definition of the **Action** element in RDL 2003/10.

```
<xsd:complexType name="ActionType">
  <xsd:all>
    <xsd:element name="Hyperlink" type="xsd:string" minOccurs="0" />
    <xsd:element name="Drillthrough" type="DrillthroughType" minOccurs="0" />
    <xsd:element name="BookmarkLink" type="xsd:string" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
```

The following is the XML Schema definition of the **Action** element in RDL 2005/01.

```
<xsd:complexType name="ActionType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hyperlink" type="xsd:string" minOccurs="0" />
    <xsd:element name="Drillthrough" type="DrillthroughType" minOccurs="0" />
    <xsd:element name="BookmarkLink" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **Action** element in RDL 2008/01.

```
<xsd:complexType name="ActionType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hyperlink" type="xsd:string" minOccurs="0" />
    <xsd:element name="Drillthrough" type="DrillthroughType" minOccurs="0" />
    <xsd:element name="BookmarkLink" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **Action** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="ActionType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hyperlink" type="xsd:string" minOccurs="0" />
    <xsd:element name="Drillthrough" type="DrillthroughType" minOccurs="0" />
    <xsd:element name="BookmarkLink" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.287.1 Action.BookmarkLink

The **Action.BookmarkLink** element specifies the ID of the bookmark that is located in a report to go to when the containing report item for this action is clicked. This element is optional.

The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If a bookmark with the specified ID is not found, the link MUST NOT be

included in the report. If the bookmark is hidden, the link MUST go to the start of the page on which the bookmark is located. If multiple bookmarks with this ID are found, the link MUST go to the bookmark that appears first in the report.

Following is the parent element of the **Action.BookmarkLink** element.

Parent elements
<a href="#">Action</a>

The following is the XML Schema definition of the **Action.BookmarkLink** element.

```
<xsd:element name="BookmarkLink" type="xsd:string" minOccurs="0">
```

### 2.287.2 Action.Drillthrough

The **Action.Drillthrough** element specifies a **drillthrough report** to be executed and viewed by clicking the containing report item for this action. This element is optional and is of type [Drillthrough](#).

Following is the parent element of the **Action.Drillthrough** element.

Parent elements
<a href="#">Action</a>

The following is the XML Schema definition of the **Action.Drillthrough** element.

```
<xsd:element name="Drillthrough" type="DrillthroughType" minOccurs="0">
```

### 2.287.3 Action.Hyperlink

The **Action.Hyperlink** element specifies a hyperlink that is viewed by clicking the containing report item for this action. This element is optional. The value of this element MUST be an [RdlURL](#).

Following is the parent element of the **Action.Hyperlink** element.

Parent elements
<a href="#">Action</a>

The following is the XML Schema definition of the **Action.Hyperlink** element.

```
<xsd:element name="Hyperlink" type="xsd:string" minOccurs="0">
```

### 2.287.4 Action.Label

**Applies to** [RDL 2005/01](#)

The **Action.Label** element is ignored.

Following is the parent element of the **Action.Label** element.



<b>Parent elements</b>
<a href="#">Action</a>

The following is the XML Schema definition of the **Action.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0">
```

## 2.288 Drillthrough

The **Drillthrough** element specifies properties for a [Report](#) that are executed by clicking the containing report item that is associated with an [Action.Drillthrough](#) element.

The following are the parent and child elements of the **Drillthrough** element.

<b>Parent elements</b>
<a href="#">Action</a>

<b>Child elements</b>
<a href="#">Drillthrough.Parameters</a>
<a href="#">Drillthrough.ReportName</a>

The following is the XML Schema definition of the **Drillthrough** element.

```
<xsd:complexType name="DrillthroughType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportName" type="xsd:string" />
    <xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.288.1 Drillthrough.Parameters

The **Drillthrough.Parameters** element specifies the parameters to be passed to a **drillthrough report**. This element is optional and is of type [Parameters](#).

Following is the parent element of the **Drillthrough.Parameters** element.

<b>Parent elements</b>
<a href="#">Drillthrough</a>

The following is the XML Schema definition of the **Drillthrough.Parameters** element.

```
<xsd:element name="Parameters" type="ParametersType" minOccurs="0">
```

## 2.288.2 Drillthrough.ReportName

The **Drillthrough.ReportName** element specifies the location of the [Report](#) to use as a **drillthrough report**. This element MUST be specified.

The location MUST specify, as a non-empty string (ignoring white space), either the absolute path (such as "/salesreports/orderdetails") or the relative path (such as "orderdetails") to a report on the same server. Relative paths start in the same folder as the main report.

Following is the parent element of the **Drillthrough.ReportName** element.

Parent elements
<a href="#">Drillthrough</a>

The following is the XML Schema definition of the **Drillthrough.ReportName** element.

```
<xsd:element name="ReportName" type="xsd:string">
```

## 2.289 Custom

**Applies to** [RDL 2003/10](#)

The **Custom** element specifies properties for which report design tools can pass information to custom report renderers and **custom report items**. The contents of the **Custom** element MUST be valid XML [\[XML10/5\]](#).

Following are the parent elements of the **Custom** element.

Parent elements
<a href="#">Report</a>
<a href="#">Image</a>
<a href="#">Line</a>
<a href="#">Rectangle</a>
<a href="#">Subreport</a>
<a href="#">Textbox</a>
<a href="#">List</a>
<a href="#">Matrix</a>
<a href="#">Table</a>
<a href="#">Grouping</a>
<a href="#">Chart</a>

The following is the XML Schema definition of the **Custom** element.

```
<xsd:complexType name="CustomType">  
  <xsd:sequence>  
    <xsd:any processContents="skip" minOccurs="0" maxOccurs="unbounded" />  
  </xsd:sequence>
```

</xsd:complexType>

## 2.290 CustomProperties

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomProperties** element specifies properties about which report design tools can pass information to custom report renderers and custom report items.

The following are the parent and child elements of the **CustomProperties** element.

Parent elements
<a href="#">Report</a>
<a href="#">Image</a>
<a href="#">Line</a>
<a href="#">Rectangle</a>
<a href="#">Subreport</a>
<a href="#">Textbox</a>
<a href="#">Tablix</a>
<a href="#">TablixMember</a>
<a href="#">List</a>
<a href="#">Matrix</a>
<a href="#">Table</a>
<a href="#">Grouping</a>
<a href="#">Chart</a>
<a href="#">ChartAxis</a>
<a href="#">ChartSeries</a>
<a href="#">ChartDataPoint</a>
<a href="#">ChartEmptyPoints</a>
<a href="#">ChartMember</a>
<a href="#">GaugePanel</a>
<a href="#">Map</a>
<a href="#">CustomReportItem</a>
<a href="#">DataMember</a>
<a href="#">DataGrouping</a>

<b>Child elements</b>
-----------------------

<a href="#">CustomProperties.CustomProperty</a>
---

The following is the XML Schema definition of the **CustomProperties** element in RDL 2005/01 and RDL 2008/01.

```
<xsd:complexType name="CustomPropertiesType">
  <xsd:sequence>
    <xsd:element name="CustomProperty" type="CustomPropertyType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **CustomProperties** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="CustomPropertiesType">
  <xsd:sequence>
    <xsd:element name="CustomProperty" type="CustomPropertyType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.290.1 CustomProperties.CustomProperty

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomProperties.CustomProperty** element specifies a **custom property** for a [Report](#), report item, or member. This element is of type [CustomProperty](#). This element MUST be specified at least once within a [CustomProperties](#) collection.

Following is the parent element of the **CustomProperties.CustomProperty** element.

<b>Parent elements</b>
------------------------

<a href="#">CustomProperties</a>
----------------------------------

The following is the XML Schema definition of the **CustomProperties.CustomProperty** element.

```
<xsd:element name="CustomProperty" type="CustomPropertyType" maxOccurs="unbounded">
```

## 2.291 CustomProperty

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomProperty** element specifies the contents of a **CustomProperty** that is passed to rendering and to [CustomReportItem](#) components. This element MUST be specified at least once within a [CustomProperties](#) collection.

The following are the parent and child elements of the **CustomProperty** element.

<b>Parent elements</b>
CustomProperties

<b>Child elements</b>
<a href="#">CustomProperty.Name</a>
<a href="#">CustomProperty.Value</a>

The following is the XML Schema definition of the **CustomProperty** element in RDL 2005/01 and RDL 2008/01.

```
<xsd:complexType name="CustomPropertyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" />
    <xsd:element name="Value" type="xsd:string" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **CustomProperty** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="CustomPropertyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" />
    <xsd:element name="Value" type="xsd:string" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.291.1 CustomProperty.Name

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomProperty.Name** element specifies the name of a [CustomProperty](#). The value of this element MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. This element MUST NOT have a **null value** or a value that is the same as another **CustomProperty.Name**. The **CustomProperty.Name** element MUST be specified.

Following is the parent element of the **CustomProperty.Name** element.

<b>Parent elements</b>
CustomProperty

The following is the XML Schema definition of the **CustomProperty.Name** element.

```
<xsd:element name="Name" type="xsd:string">
```

## 2.291.2 CustomProperty.Value

Applies to [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **CustomProperty.Value** element specifies the value of a [CustomProperty](#). The value of this element MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **Variant**. This element MUST be specified.

Following is the parent element of the **CustomProperty.Value** element.

Parent elements
<a href="#">CustomProperty</a>

The following is the XML Schema definition of the **CustomProperty.Value** element.

```
<xsd:element name="Value" type="xsd:string">
```

## 2.292 PageBreak

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PageBreak** element specifies page break behavior for a rectangle, data region, or group.

The following are the parent and child elements of the **PageBreak** element.

Parent elements
<a href="#">Rectangle</a>
<a href="#">Group</a>
<a href="#">Chart</a>
<a href="#">Tablix</a>
<a href="#">GaugePanel</a>
<a href="#">Map</a>

Child elements
<a href="#">PageBreak.BreakLocation</a>
<a href="#">PageBreak.Disabled</a>
<a href="#">PageBreak.ResetPageNumber</a>

The following is the XML Schema definition of the **PageBreak** element in [RDL 2003/10](#), [RDL 2005/01](#), and [RDL 2008/01](#).

```
<xsd:complexType name="PageBreakType">  
  <xsd:choice minOccurs="1" maxOccurs="unbounded">  
    <xsd:element name="BreakLocation" minOccurs="1">  
      <xsd:simpleType>  
        <xsd:restriction base="xsd:string">  
          <xsd:enumeration value="None" />  
        </xsd:restriction>  
      </xsd:simpleType>  
    </xsd:element>  
  </xsd:choice>  
</xsd:complexType>
```

```

        <xsd:enumeration value="Start" />
        <xsd:enumeration value="End" />
        <xsd:enumeration value="StartAndEnd" />
        <xsd:enumeration value="Between" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

Following is the XML Schema definition of the **PageBreak** element in RDL 2010/01 and RDL 2016/01.

```

<xsd:complexType name="PageBreakType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Disabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="ResetPageNumber" type="xsd:string" minOccurs="0" />
    <xsd:element name="BreakLocation" minOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Start" />
          <xsd:enumeration value="End" />
          <xsd:enumeration value="StartAndEnd" />
          <xsd:enumeration value="Between" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.292.1 PageBreak.BreakLocation

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **PageBreak.BreakLocation** element specifies where a page break occurs in a group or a report item. This element **MUST** be specified. The value of this element **MUST** be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is one of the following:

**None:** Specifies that the **PageBreak.BreakLocation** element is ignored.

**Start:** Specifies that there is a page break before a report item or each instance of a group.

**End:** Specifies that there is a page break after a report item or each instance of a group.

**StartAndEnd:** Specifies that there is a page break both before and after a report item or each instance of a group.

**Between:** Specifies that there is a page break between each instance of a group. This value cannot apply to report items.

Following is the parent element of the **PageBreak.BreakLocation** element.

<b>Parent elements</b>
<a href="#">PageBreak</a>

The following is the XML Schema definition of the **PageBreak.BreakLocation** element.

```

<xsd:element name="BreakLocation" minOccurs="1">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None" />
      <xsd:enumeration value="Start" />
      <xsd:enumeration value="End" />
      <xsd:enumeration value="StartAndEnd" />
      <xsd:enumeration value="Between" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

```

## 2.292.2 PageBreak.Disabled

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **PageBreak.Disabled** element specifies whether the properties of a [PageBreak](#) element ([PageBreak.BreakLocation](#) and [PageBreak.ResetPageNumber](#)) is ignored.

The **PageBreak.Disabled** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **PageBreak.Disabled** element.

Parent elements
<a href="#">PageBreak</a>

The following is the XML Schema definition of the **PageBreak.Disabled** element.

```

<xsd:element name="Disabled" type="xsd:string" minOccurs="0" />

```

## 2.292.3 PageBreak.ResetPageNumber

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

The **PageBreak.ResetPageNumber** element specifies that the values for page numbering for paginated pages are reset. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2) or an expression that evaluates to a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **PageBreak.ResetPageNumber** element.

Parent elements
<a href="#">PageBreak</a>

The following is the XML Schema definition of the **PageBreak.ResetPageNumber** element.

```

<xsd:element name="ResetPageNumber" type="xsd:string" minOccurs="0" />

```

## 2.293 Style

The **Style** element specifies appearance information for a report item.



The following are the parent and child elements of the **Style** element.

Parent elements
<a href="#">Line</a>
<a href="#">Rectangle</a>
<a href="#">Textbox</a>
<a href="#">Paragraph</a>
<a href="#">TextRun</a>
<a href="#">Image</a>
<a href="#">SubReport</a>
<a href="#">Tablix</a>
<a href="#">Chart</a>
<a href="#">Body</a>
<a href="#">PageSection</a>
<a href="#">Page</a>
<a href="#">ChartAxis</a>
<a href="#">ChartBorderSkin</a>
<a href="#">ChartTitle</a>
<a href="#">ChartLegend</a>
<a href="#">ChartLegendTitle</a>
<a href="#">ChartArea</a>
<a href="#">ChartAxisTitle</a>
<a href="#">ChartAxisScaleBreak</a>
<a href="#">ChartSeries</a>
<a href="#">ChartDataPoint</a>
<a href="#">ChartEmptyPoints</a>
<a href="#">ChartDataLabel</a>
<a href="#">ChartMarker</a>
<a href="#">ChartGridLines</a>
<a href="#">ChartTickMarks</a>
<a href="#">ChartStripLine</a>
<a href="#">LinearGauge</a>
<a href="#">RadialGauge</a>
<a href="#">GaugePanel</a>

<b>Parent elements</b>
<a href="#">LinearPointer</a>
<a href="#">RadialPointer</a>
<a href="#">ScaleRange</a>
<a href="#">LinearScale</a>
<a href="#">RadialScale</a>
<a href="#">Thermometer</a>
<a href="#">PointerCap</a>
<a href="#">ScaleLabels</a>
<a href="#">CustomLabel</a>
<a href="#">TickMarkStyle</a>
<a href="#">GaugeTickMarks</a>
<a href="#">ScalePin</a>
<a href="#">PinLabel</a>
<a href="#">BackFrame</a>
<a href="#">FrameBackground</a>
<a href="#">ChartLegendColumn</a>
<a href="#">ChartLegendColumnHeader</a>
<a href="#">ChartLegendCustomItem</a>
<a href="#">ChartLegendCustomItemCell</a>
<a href="#">Axis</a>
<a href="#">DataLabel</a>
<a href="#">DataPoint</a>
<a href="#">MajorGridLines</a>
<a href="#">Marker</a>
<a href="#">MinorGridLines</a>
<a href="#">PlotArea</a>
<a href="#">Title</a>
<a href="#">GaugeLabel</a>
<a href="#">NumericIndicator</a>
<a href="#">StateIndicator</a>
<a href="#">IndicatorState</a>
<a href="#">CustomReportItem</a>

<b>Parent elements</b>
<a href="#">List</a>
<a href="#">Matrix</a>
<a href="#">Table</a>
<a href="#">Subtotal</a>
<a href="#">Map</a>
<a href="#">MapBorderSkin</a>
<a href="#">MapColorScale</a>
<a href="#">MapColorScaleTitle</a>
<a href="#">MapDistanceScale</a>
<a href="#">MapLegend</a>
<a href="#">MapLegendTitle</a>
<a href="#">MapLineTemplate</a>
<a href="#">MapMarkerTemplate</a>
<a href="#">MapPolygonTemplate</a>
<a href="#">MapTitle</a>
<a href="#">MapViewport</a>

<b>Child elements</b>
<a href="#">Style.BackgroundColor</a>
<a href="#">Style.BackgroundGradientEndColor</a>
<a href="#">Style.BackgroundGradientType</a>
<a href="#">Style.BackgroundHatchType</a>
<a href="#">Style.BackgroundImage</a>
<a href="#">Style.Border</a>
<a href="#">Style.BottomBorder</a>
<a href="#">Style.Calendar</a>
<a href="#">Style.Color</a>
<a href="#">Style.Direction</a>
<a href="#">Style.FontFamily</a>
<a href="#">Style.FontSize</a>
<a href="#">Style.FontStyle</a>

Child elements
<a href="#">Style.FontWeight</a>
<a href="#">Style.Format</a>
<a href="#">Style.Language</a>
<a href="#">Style.LeftBorder</a>
<a href="#">Style.LineHeight</a>
<a href="#">Style.NumeralLanguage</a>
<a href="#">Style.NumeralVariant</a>
<a href="#">Style.PaddingBottom</a>
<a href="#">Style.PaddingLeft</a>
<a href="#">Style.PaddingRight</a>
<a href="#">Style.PaddingTop</a>
<a href="#">Style.RightBorder</a>
<a href="#">Style.ShadowColor</a>
<a href="#">Style.ShadowOffset</a>
<a href="#">Style.TextAlign</a>
<a href="#">Style.TextDecoration</a>
<a href="#">Style.TextEffect</a>
<a href="#">Style.TopBorder</a>
<a href="#">Style.VerticalAlign</a>
<a href="#">Style.WritingMode</a>
<a href="#">Style.UnicodeBiDi</a>
<a href="#">Style.BorderColor</a>
<a href="#">Style.BorderStyle</a>
<a href="#">Style.BorderWidth</a>

**Applies to [RDL 2013/01](#)**

Child elements
<a href="#">Style.CurrencyLanguage</a>

The following is the XML Schema definition of the **Style** element in [RDL 2003/10](#) and [RDL 2005/01](#).

```
<xsd:complexType name="StyleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="BorderColor" type="BorderColorStyleWidthType" minOccurs="0" />
    <xsd:element name="BorderStyle" type="BorderColorStyleWidthType" minOccurs="0" />
    <xsd:element name="BorderWidth" type="BorderColorStyleWidthType" minOccurs="0" />
    <xsd:element name="BackgroundColor" type="xsd:string" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

```

<xsd:element name="BackgroundGradientType" type="xsd:string" minOccurs="0" />
<xsd:element name="BackgroundGradientEndColor" type="xsd:string" minOccurs="0" />
<xsd:element name="BackgroundHatchType" type="xsd:string" minOccurs="0" />
<xsd:element name="BackgroundImage" type="BackgroundImageType" minOccurs="0" />
<xsd:element name="FontStyle" type="xsd:string" minOccurs="0" />
<xsd:element name="FontFamily" type="xsd:string" minOccurs="0" />
<xsd:element name="FontSize" type="xsd:string" minOccurs="0" />
<xsd:element name="FontWeight" type="xsd:string" minOccurs="0" />
<xsd:element name="Format" type="xsd:string" minOccurs="0" />
<xsd:element name="TextDecoration" type="xsd:string" minOccurs="0" />
<xsd:element name="TextAlign" type="xsd:string" minOccurs="0" />
<xsd:element name="VerticalAlign" type="xsd:string" minOccurs="0" />
<xsd:element name="Color" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingLeft" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingRight" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingTop" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingBottom" type="xsd:string" minOccurs="0" />
<xsd:element name="LineHeight" type="xsd:string" minOccurs="0" />
<xsd:element name="Direction" type="xsd:string" minOccurs="0" />
<xsd:element name="WritingMode" type="xsd:string" minOccurs="0" />
<xsd:element name="Language" type="xsd:string" minOccurs="0" />
<xsd:element name="UnicodeBiDi" type="xsd:string" minOccurs="0" />
<xsd:element name="Calendar" type="xsd:string" minOccurs="0" />
<xsd:element name="NumeralLanguage" type="xsd:string" minOccurs="0" />
<xsd:element name="NumeralVariant" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **Style** element in [RDL 2008/01](#).

```

<xsd:complexType name="StyleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Border" type="BorderType" minOccurs="0" />
    <xsd:element name="TopBorder" type="BorderType" minOccurs="0" />
    <xsd:element name="BottomBorder" type="BorderType" minOccurs="0" />
    <xsd:element name="LeftBorder" type="BorderType" minOccurs="0" />
    <xsd:element name="RightBorder" type="BorderType" minOccurs="0" />
    <xsd:element name="BackgroundColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundGradientType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundGradientEndColor" type="xsd:string"
      minOccurs="0" />
    <xsd:element name="BackgroundHatchType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundImage" type="BackgroundImageType" minOccurs="0" />
    <xsd:element name="FontStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontFamily" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontWeight" type="xsd:string" minOccurs="0" />
    <xsd:element name="Format" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextDecoration" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextAlign" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextEffect" type="xsd:string" minOccurs="0" />
    <xsd:element name="VerticalAlign" type="xsd:string" minOccurs="0" />
    <xsd:element name="Color" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShadowColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShadowOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="PaddingLeft" type="xsd:string" minOccurs="0" />
    <xsd:element name="PaddingRight" type="xsd:string" minOccurs="0" />
    <xsd:element name="PaddingTop" type="xsd:string" minOccurs="0" />
    <xsd:element name="PaddingBottom" type="xsd:string" minOccurs="0" />
    <xsd:element name="LineHeight" type="xsd:string" minOccurs="0" />
    <xsd:element name="Direction" type="xsd:string" minOccurs="0" />
    <xsd:element name="WritingMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="Language" type="xsd:string" minOccurs="0" />
    <xsd:element name="UnicodeBiDi" type="xsd:string" minOccurs="0" />
    <xsd:element name="Calendar" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:element name="NumeralLanguage" type="xsd:string" minOccurs="0" />
    <xsd:element name="NumeralVariant" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **Style** element in [RDL 2010/01](#) and [RDL 2016/01](#).

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections [5.5](#), [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```

<xsd:complexType name="StyleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Border" type="BorderType" minOccurs="0" />
    <xsd:element name="TopBorder" type="BorderType" minOccurs="0" />
    <xsd:element name="BottomBorder" type="BorderType" minOccurs="0" />
    <xsd:element name="LeftBorder" type="BorderType" minOccurs="0" />
    <xsd:element name="RightBorder" type="BorderType" minOccurs="0" />
    <xsd:element name="BackgroundColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundGradientType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundGradientEndColor" type="xsd:string"
      minOccurs="0" />
    <xsd:element name="BackgroundHatchType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundImage" type="BackgroundImageType" minOccurs="0" />
    <xsd:element name="FontStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontFamily" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontWeight" type="xsd:string" minOccurs="0" />
    <xsd:element name="Format" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextDecoration" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextAlign" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextEffect" type="xsd:string" minOccurs="0" />
    <xsd:element name="VerticalAlign" type="xsd:string" minOccurs="0" />
    <xsd:element name="Color" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShadowColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShadowOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="PaddingLeft" type="xsd:string" minOccurs="0" />
    <xsd:element name="PaddingRight" type="xsd:string" minOccurs="0" />
    <xsd:element name="PaddingTop" type="xsd:string" minOccurs="0" />
    <xsd:element name="PaddingBottom" type="xsd:string" minOccurs="0" />
    <xsd:element name="LineHeight" type="xsd:string" minOccurs="0" />
    <xsd:element name="Direction" type="xsd:string" minOccurs="0" />
    <xsd:element name="WritingMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="Language" type="xsd:string" minOccurs="0" />
    <xsd:element name="UnicodeBiDi" type="xsd:string" minOccurs="0" />
    <xsd:element name="Calendar" type="xsd:string" minOccurs="0" />
    <xsd:element name="NumeralLanguage" type="xsd:string" minOccurs="0" />
    <xsd:element name="NumeralVariant" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.293.1 **Style.BackgroundColor**

The **Style.BackgroundColor** element specifies the color of the background of a report item. This element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

In [RDL 2012/01](#), **Style.BackgroundColor** is of type [StringWithValueTypeAttribute](#). If the **ValueType** attribute is set to **ThemeReference**, the value MUST be the name of a theme color. If **ValueType** is not present or is set to "Constant", the value MUST be a value that is specified as an **RdIColor**.

If the **Style.BackgroundColor** element is not present, the background of the report item is interpreted as being transparent. If the grandparent element of the **Style.BackgroundColor** element is [Line](#), [Paragraph](#), [TextRun](#), [SubReport](#), [Image](#), [ChartAxis](#), [Axis](#), [ChartAxisTitle](#), [Title](#), [ChartAxisScaleBreak](#), [ScaleLabels](#), [CustomLabel](#), or [PinLabel](#), the **Style.BackgroundColor** element is ignored.

Following is the parent element of the **Style.BackgroundColor** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.BackgroundColor** element in [RDL 2003/10](#), [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#).

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections [5.5](#), [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:element name="BackgroundColor" type="xsd:string" minOccurs="0" />
```

## 2.293.2 **Style.BackgroundGradientEndColor**

The **Style.BackgroundGradientEndColor** element specifies the end color for the background gradient of a report item. This element is optional. If this element is present, its value MUST be an [RdIColor](#) value or an expression that evaluates to an **RdIColor**.

The **Style.BackgroundGradientEndColor** element is ignored if the sibling element [Style.BackgroundGradientType](#) is interpreted as "None". If the grandparent element of the **Style.BackgroundGradientEndColor** element is a report item and not a Chart, the **Style.BackgroundGradientEndColor** element is ignored.

If the grandparent element of this element is [ChartAxis](#), [Axis](#), [ChartAxisTitle](#), [Title](#), [ChartAxisScaleBreak](#), [MajorGridLines](#), [MinorGridLines](#), [ChartGridLines](#), [ChartTickMarks](#), [LinearGauge](#), [RadialGauge](#), [ScaleRange](#), [ScaleLabels](#), [CustomLabel](#), [CustomLabel.TickMarkStyle](#), or [PinLabel](#), the **Style.BackgroundGradientEndColor** element is ignored.

Following is the parent element of the **Style.BackgroundGradientEndColor** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.BackgroundGradientEndColor** element.

```
<xsd:element name="BackgroundGradientEndColor" type="xsd:string" minOccurs="0" />
```

## 2.293.3 **Style.BackgroundGradientType**

The **Style.BackgroundGradientType** element specifies the type of background gradient of a report item. This element is optional. If this element is present, its value MUST be a [String](#)

([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST evaluate to one of the following:

**Default:** Specifies the default background gradient, which is treated the same as "None".

**None:** Specifies that there is no background gradient.

**LeftRight:** Specifies a horizontal background gradient.

**TopBottom:** Specifies a vertical background gradient.

**Center:** Specifies a circular background gradient.

**DiagonalLeft:** Specifies a diagonal gradient that goes from top-left to bottom-right.

**DiagonalRight:** Specifies a diagonal gradient that goes from top-right to bottom-left.

**HorizontalCenter:** Specifies a horizontal gradient that goes out from the center to both the top and the bottom.

**VerticalCenter:** Specifies a vertical gradient that goes out from the center to both the left and the right.

If the **Style.BackgroundGradientType** element is not present, its value is interpreted as "None". If the grandparent element of this element is a **ReportItem** and not a [Chart](#), the **Style.BackgroundGradientType** element is ignored.

If the grandparent element of this element is [ChartAxis](#), [Axis](#), [Title](#), [ChartAxisTitle](#), [ChartAxisScaleBreak](#), [ChartMarker](#), [Marker](#), [ChartGridLines](#), [MajorGridLines](#), [MinorGridLines](#), [ChartTickMarks](#), [LinearGauge](#), [RadialGauge](#), [ScaleRange](#), [ScaleLabels](#), [CustomLabel](#), or [PinLabel](#), the **Style.BackgroundGradientType** element is ignored.

Following is the parent element of the **Style.BackgroundGradientType** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.BackgroundGradientType** element.

```
<xsd:element name="BackgroundGradientType" type="xsd:string" minOccurs="0" />
```

## 2.293.4 **Style.BackgroundHatchType**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Style.BackgroundHatchType** element specifies the hatching pattern of a report item. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies the default background hatching pattern, which is treated the same as "None".

**None:** Specifies that there is no background hatching.

**BackwardDiagonal:** Specifies a backward-diagonal hatching pattern.

**Cross:** Specifies a cross-hatching pattern.



**DarkDownwardDiagonal:** Specifies a dark downward-diagonal hatching pattern.

**DarkHorizontal:** Specifies a dark horizontal hatching pattern.

**DarkUpwardDiagonal:** Specifies a dark upward-diagonal hatching pattern.

**DarkVertical:** Specifies a dark vertical hatching pattern.

**DashedDownwardDiagonal:** Specifies a dashed downward-diagonal hatching pattern.

**DashedHorizontal:** Specifies a dashed horizontal hatching pattern.

**DashedUpwardDiagonal:** Specifies a dashed upward-diagonal hatching pattern.

**DashedVertical:** Specifies a dashed vertical hatching pattern.

**DiagonalBrick:** Specifies a diagonal brick hatching pattern.

**DiagonalCross:** Specifies a diagonal cross-hatching pattern.

**Divot:** Specifies a divot hatching pattern.

**DottedDiamond:** Specifies a dotted diamond hatching pattern.

**DottedGrid:** Specifies a dotted grid hatching pattern.

**ForwardDiagonal:** Specifies a forward-diagonal hatching pattern.

**Horizontal:** Specifies a horizontal hatching pattern.

**HorizontalBrick:** Specifies a horizontal brick hatching pattern.

**LargeCheckerBoard:** Specifies a large checkerboard hatching pattern.

**LargeConfetti:** Specifies a large confetti hatching pattern.

**LargeGrid:** Specifies a large grid hatching pattern.

**LightDownwardDiagonal:** Specifies a light downward-diagonal hatching pattern.

**LightHorizontal:** Specifies a light horizontal hatching pattern.

**LightUpwardDiagonal:** Specifies a light upward-diagonal hatching pattern.

**LightVertical:** Specifies a light vertical hatching pattern.

**NarrowHorizontal:** Specifies a narrow horizontal hatching pattern.

**NarrowVertical:** Specifies a narrow vertical hatching pattern.

**OutlinedDiamond:** Specifies an outlined diamond hatching pattern.

**Percent05:** Specifies a 5-percent hatching pattern.

**Percent10:** Specifies a 10-percent hatching pattern.

**Percent20:** Specifies a 20-percent hatching pattern.

**Percent25:** Specifies a 25-percent hatching pattern.

**Percent30:** Specifies a 30-percent hatching pattern.

**Percent40:** Specifies a 40-percent hatching pattern.

**Percent50:** Specifies a 50-percent hatching pattern.

**Percent60:** Specifies a 60-percent hatching pattern.

**Percent70:** Specifies a 70-percent hatching pattern.

**Percent75:** Specifies a 75-percent hatching pattern.

**Percent80:** Specifies an 80-percent hatching pattern.

**Percent90:** Specifies a 90-percent hatching pattern.

**Plaid:** Specifies a plaid hatching pattern.

**Shingle:** Specifies a shingle hatching pattern.

**SmallCheckerBoard:** Specifies a small checkerboard hatching pattern.

**SmallConfetti:** Specifies a small confetti hatching pattern.

**SmallGrid:** Specifies a small grid hatching pattern.

**SolidDiamond:** Specifies a solid diamond hatching pattern.

**Sphere:** Specifies a sphere hatching pattern.

**Trellis:** Specifies a trellis hatching pattern.

**Vertical:** Specifies a vertical hatching pattern.

**Wave:** Specifies a wave hatching pattern.

**Weave:** Specifies a weave hatching pattern.

**WideDownwardDiagonal:** Specifies a wide downward-diagonal hatching pattern.

**WideUpwardDiagonal:** Specifies a wide upward-diagonal hatching pattern.

**ZigZag:** Specifies a zigzag hatching pattern.

If the **Style.BackgroundHatchType** element is not present, its value is interpreted as "None". If the grandparent element of this element is [ReportItems](#), [ChartTitle](#), [Title](#), [ChartAxis](#), [Axis](#), [ChartAxisTitle](#), [ChartAxisScaleBreak](#), [ChartMarker](#), [Marker](#), [ChartGridLines](#), [MajorGridLines](#), [MinorGridLines](#), [ChartTickMarks](#), [LinearGauge](#), [RadialGauge](#), [ScaleRange](#), [ScaleLabels](#), [CustomLabel](#), or [PinLabel](#), the **Style.BackgroundHatchType** element is ignored.

Following is the parent element of the **Style.BackgroundHatchType** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.BackgroundHatchType** element.

```
<xsd:element name="BackgroundHatchType" type="xsd:string" minOccurs="0" />
```

## 2.293.5 Style.BackgroundImage

The **Style.BackgroundImage** element specifies the [BackgroundImage](#) for a report item. This element is optional. If this element is not present, there **MUST NOT** be a **background image** for the report item. The **Style.BackgroundImage** element is of type **BackgroundImage**.

Following is the parent element of the **Style.BackgroundImage** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.BackgroundImage** element.

```
<xsd:element name="BackgroundImage" type="BackgroundImageType" minOccurs="0" />
```

## 2.293.6 Style.Border

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Style.Border** element specifies the default border properties for a report item. This element is optional and is of type [Border](#).

Following is the parent element of the **Style.Border** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.Border** element.

```
<xsd:element name="Border" type="BorderType" minOccurs="0" />
```

## 2.293.7 Style.BorderColor

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Style.BorderColor** element specifies the color of a border. This element is optional and is of type [BorderColor](#).

Following is the parent element of the **Style.BorderColor** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.BorderColor** element.

```
<xsd:element name="BorderColor" type="BorderColorStyleWidthType" minOccurs="0" />
```

## 2.293.8 Style.BorderStyle

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Style.BorderStyle** element specifies the style of a border. This element is optional and is of type [BorderStyle](#).

Following is the parent element of the **Style.BorderStyle** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.BorderStyle** element.

```
<xsd:element name="BorderStyle" type="BorderColorStyleWidthType" minOccurs="0" />
```

## 2.293.9 Style.BorderWidth

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **Style.BorderWidth** element specifies the width of a border. This element is optional and is of type [BorderWidth](#).

Following is the parent element of the **Style.BorderWidth** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.BorderWidth** element.

```
<xsd:element name="BorderWidth" type="BorderColorStyleWidthType" minOccurs="0" />
```

## 2.293.10 Style.BottomBorder

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Style.BottomBorder** element specifies properties for the bottom border of a report item. These properties override the corresponding value in the default sibling [Style.Border](#) element. The **Style.BottomBorder** element is optional and is of type [Border](#).

If the grandparent element of this element is not [Line](#), [Rectangle](#), [Textbox](#), [Image](#), [SubReport](#), [PageSection](#), [Tablix](#), [Chart](#), [GaugePanel](#), or [Page](#), the **Style.BottomBorder** element is ignored.

Following is the parent element of the **Style.BottomBorder** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.BottomBorder** element.

```
<xsd:element name="BottomBorder" type="BorderType" minOccurs="0" />
```

## 2.293.11 Style.Calendar

The **Style.Calendar** element specifies the calendar to use for formatting dates. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies the default calendar for the language of the report item.

**Gregorian:** Specifies the Gregorian calendar standard.

**GregorianArabic:** Specifies the Gregorian Arabic calendar standard.

**GregorianMiddleEastFrench:** Specifies the Gregorian Middle East French calendar standard.

**GregorianTransliteratedEnglish:** Specifies the Gregorian Transliterated English calendar standard.

**GregorianTransliteratedFrench:** Specifies the Gregorian Transliterated French calendar standard.

**GregorianUSEnglish:** Specifies the Gregorian US English calendar standard.

**Hebrew:** Specifies the Hebrew calendar standard.

**Hijri:** Specifies the Hijri calendar standard.

**Japanese:** Specifies the Japanese calendar standard.

**Korean:** Specifies the Korean calendar standard.

**Taiwan:** Specifies the Taiwan calendar standard.

**ThaiBuddhist:** Specifies the Thai Buddhist calendar standard.

The value of the **Style.Calendar** element MUST be compatible in the .NET Framework with the **Language** setting. If the **Style.Calendar** element is not present, its value is interpreted as "Default".

If the grandparent element of the **Style.Calendar** element is not [TextRun](#), [Chart](#), [Textbox](#), [LinearGauge](#), or [RadialGauge](#), the **Style.Calendar** element is ignored.

Following is the parent element of the **Style.Calendar** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.Calendar** element.

```
<xsd:element name="Calendar" type="xsd:string" minOccurs="0" />
```

## 2.293.12 Style.Color

The **Style.Color** element specifies the foreground color of a report item. This element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

In [RDL 2012/01](#), **Style.Color** is of type [StringWithValueTypeAttribute](#). If the **ValueType** attribute is set to **ThemeReference**, the value MUST be the name of a theme color. If **ValueType** is not present or is set to "Constant", the value MUST be a value that is specified as an **RdlColor**.

If the **Style.Color** element is not present, the foreground color of the report item is interpreted as "Black", except within the [ChartDataPoint](#) and [ChartSeries](#) elements. Within these elements, the value of the **Style.Color** element is interpreted to be the same as those that are specified for the chart's **color palette** (specified by the [Chart.Palette](#) element).

If the grandparent element of the **Style.Color** element is not [TextRun](#), [Textbox](#), [Subtotal](#), [ChartTitle](#), [ChartLegend](#), [ChartLegendTitle](#), [ChartAxis](#), [Axis](#), [ChartAxisTitle](#), [Title](#), [ChartEmptyPoints](#), [ChartDataLabel](#), [ChartStripLine](#), [ScaleLabels](#), [CustomLabel](#), or [PinLabel](#), the **Style.Color** element is ignored.

Following is the parent element of the **Style.Color** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.Color** element in [RDL 2003/10](#), [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#).

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections [5.5](#), [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:element name="Color" type="xsd:string" minOccurs="0" />
```

## 2.293.13 Style.CurrencyLanguage

**Applies to** [RDL 2013/01](#)

The **Style.CurrencyLanguage** element specifies the locale that is used for locale-dependent currency formatting in a report item. This element is optional. If this element is present, its value MUST be a [ReportLanguage](#) value.

Following is the parent element of the **Style.CurrencyLanguage** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.CurrencyLanguage** element.

```
<xsd:element name="CurrencyLanguage" type="xsd:string" />
```

## 2.293.14 Style.Direction

The **Style.Direction** element specifies whether text within a report item is written left-to-right or right-to-left. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies the default text direction, which is the same as "LTR".

**LTR:** Specifies that text is written left-to-right.

**RTL**: Specifies that the text is written right-to-left.

The **Style.Direction** element MUST NOT affect the alignment of text unless the peer [Style.TextAlign](#) element is set to "General". If the **Style.Direction** element is not present, its value is interpreted as "Default".

If the grandparent element of the **Style.Direction** element is not [TextRun](#), [SubReport](#), [Chart](#), [LinearGauge](#), or [RadialGauge](#), the **Style.Direction** element is ignored.

Following is the parent element for the **Style.Direction** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.Direction** element.

```
<xsd:element name="Direction" type="xsd:string" minOccurs="0" />
```

### 2.293.15 **Style.FontFamily**

The **Style.FontFamily** element specifies the name of the font family for text within a report item. This element is optional.

If the **Style.FontFamily** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, its value is interpreted as the value defined in [Report.DefaultFontFamily](#). If the value of this element is not a recognized font family, its value is interpreted as the value defined in **Report.DefaultFontFamily**.

In [RDL 2012/01](#), **Style.FontFamily** is of type [StringWithValueTypeAttribute](#). If the **ValueType** attribute is set to **ThemeReference**, the value MUST be the name of a theme font. If **ValueType** is not present or is set to "Constant", the value is interpreted as the name of a font family.

If the grandparent element of the **Style.FontFamily** element is not [TextRun](#), [Chart](#), [Textbox](#), [Subtotal](#), [ChartTitle](#), [ChartLegend](#), [ChartLegendTitle](#), [ChartAxis](#), [Axis](#), [ChartAxisTitle](#), [Title](#), [ChartDataLabel](#), [ChartStripLine](#), [ScaleLabels](#), [CustomLabel](#), or [PinLabel](#), the **Style.FontFamily** element is ignored.

Following is the parent element of the **Style.FontFamily** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.FontFamily** element.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections [5.5](#), [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:element name="FontFamily" type="xsd:string" minOccurs="0" />
```

## 2.293.16 Style.FontSize

The **Style.FontSize** element specifies the font size for text in a report item. This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**. If the **Style.FontSize** element is not present, its value is interpreted as 10pt. The value of the **Style.FontSize** element MUST NOT be a size that is greater than 200 pt or the equivalent. The value of this element MUST NOT be a size that is less than 1 pt or the equivalent.

If the grandparent element of the **Style.FontSize** element is not [TextRun](#), [Chart](#), [Textbox](#), [Subtotal](#), [ChartTitle](#), [ChartLegend](#), [ChartLegendTitle](#), [ChartAxis](#), [Axis](#), [ChartAxisTitle](#), [Title](#), [ChartDataLabel](#), [ChartStripLine](#), [ScaleLabels](#), [CustomLabel](#), or [PinLabel](#), this element is ignored.

Following is the parent element of the **Style.FontSize** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.FontSize** element.

```
<xsd:element name="FontSize" type="xsd:string" minOccurs="0" />
```

## 2.293.17 Style.FontStyle

The **Style.FontStyle** element specifies the font size for text in a report item. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies the default font style, which is the same as "Normal".

**Normal:** Specifies that the font is not italicized.

**Italic:** Specifies that the font is italicized.

If the **Style.FontStyle** element is not present, its value is interpreted as "Default".

If the grandparent element of the **Style.FontStyle** element is not [TextRun](#), [Chart](#), [Textbox](#), [TextRun](#), [ChartTitle](#), [ChartLegend](#), [ChartLegendTitle](#), [ChartAxis](#), [Axis](#), [ChartAxisTitle](#), [Title](#), [ChartDataLabel](#), [ChartStripLine](#), [ScaleLabels](#), [CustomLabel](#), or [PinLabel](#), the **Style.FontStyle** element is ignored.

Following is the parent element of the **Style.FontStyle** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.FontStyle** element.

```
<xsd:element name="FontStyle" type="xsd:string" minOccurs="0" />
```

## 2.293.18 Style.FontWeight

The **Style.FontWeight** element specifies the thickness of font for text within a report item. This element is optional. If the **Style.FontWeight** element is present, its value MUST be a [String](#)



([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies the default font thickness, which is the same as "Normal".

**Thin:** Specifies a font thickness that matches cascading style sheets (CSS) font weight 100.

**ExtraLight:** Specifies a font thickness that matches CSS font weight 200.

**Light:** Specifies a font thickness that matches CSS font weight 300.

**Normal:** Specifies a font thickness that matches CSS font weight 400.

**Medium:** Specifies a font thickness that matches CSS font weight 500.

**SemiBold:** Specifies a font thickness that matches CSS font weight 600.

**Bold:** Specifies a font thickness that matches CSS font weight 700.

**ExtraBold:** Specifies a font thickness that matches CSS font weight 800.

**Heavy:** Specifies a font thickness that matches CSS font weight 900.

If the grandparent element of the **Style.FontWeight** element is not [TextRun](#), [Chart](#), [Textbox](#), [Subtotal](#), [ChartTitle](#), [ChartLegend](#), [ChartLegendTitle](#), [ChartAxis](#), [Axis](#), [ChartAxisTitle](#), [Title](#), [ChartDataLabel](#), [ChartStripLine](#), [ScaleLabels](#), [CustomLabel](#), or [PinLabel](#), the **Style.FontWeight** element is ignored.

If the **Style.FontWeight** element is not present, its value is interpreted as "Default".

Following is the parent element of the **Style.FontWeight** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.FontWeight** element.

```
<xsd:element name="FontWeight" type="xsd:string" minOccurs="0" />
```

## 2.293.19 Style.Format

The **Style.Format** element specifies a formatting code that is used when the numeric value in a report item is formatted. This element is optional.

If the **Style.Format** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, no formatting exists for the associated text to which the **Style.Format** element applies.

If the **Style.Format** element has locale-dependent currency formatting or locale-dependent date formatting, its value MUST be based on the language setting for the report item. If the grandparent element of the **Style.Format** element is not [TextRun](#), [Chart](#), [Textbox](#), [Subtotal](#), [ChartTitle](#), [ChartLegend](#), [ChartLegendTitle](#), [ChartAxis](#), [Axis](#), [ChartAxisTitle](#), [Title](#), [ChartDataLabel](#), [ChartStripLine](#), or [ScaleLabels](#), the **Style.Format** element is ignored.

Following is the parent element of the **Style.Format** element.

<b>Parent elements</b>
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.Format** element.

```
<xsd:element name="Format" type="xsd:string" minOccurs="0" />
```

### 2.293.20 **Style.Language**

The **Style.Language** element specifies the primary language of the text in a report item. This element is optional.

If the **Style.Language** element is present, its value MUST be a [ReportLanguage](#) value. If the **Style.Language** element is not present, its value MUST be the same as that of the **Report.Language** element.

For text-formatting operations, the **Style.Language** element MUST apply only to the following:

- [Textbox.Value](#)
- [DataLabel.Value](#)
- [ChartMember.Label](#)
- [TextRun.Value](#)
- [DataValue.Value](#)

If the grandparent element of the **Style.Language** element is not [TextRun](#), [Chart](#), [Textbox](#), [Body](#), [LinearGauge](#), or [RadialGauge](#), the **Style.Language** element is ignored.

Following is the parent element for the **Style.Language** element.

<b>Parent elements</b>
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.Language** element.

```
<xsd:element name="Language" type="xsd:string" minOccurs="0" />
```

### 2.293.21 **Style.LeftBorder**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Style.LeftBorder** element specifies properties for the left border of a report item. These properties override the corresponding value in the default sibling [Style.Border](#) element. The **Style.LeftBorder** element is optional and is of type [Border](#).

If the grandparent element of this element is not [Line](#), [Rectangle](#), [Textbox](#), [Image](#), [SubReport](#), [PageSection](#), [Tablix](#), [Chart](#), [GaugePanel](#), or [Page](#), the **Style.LeftBorder** element is ignored.

Following is the parent element of the **Style.LeftBorder** element.

<b>Parent elements</b>
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.LeftBorder** element.

```
<xsd:element name="LeftBorder" type="BorderType" minOccurs="0" />
```

## 2.293.22 Style.LineHeight

The **Style.LineHeight** element specifies the height of a line of text in a report item. This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**.

If the **Style.LineHeight** element is not present, its value MUST be determined by the particular renderer that renders its associated text based on the [Style.FontSize](#) element for the text. The value of the **Style.LineHeight** element MUST NOT be a size that is greater than 1000 pt or the equivalent. The value of this element MUST NOT be a size that is less than 1 pt or the equivalent.

If the grandparent element of the **Style.LineHeight** element is not [Paragraph](#), [Textbox](#) or [Subtotal](#), the **Style.LineHeight** element is ignored.

Following is the parent element of the **Style.LineHeight** element.

<b>Parent elements</b>
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.LineHeight** element.

```
<xsd:element name="LineHeight" type="xsd:string" minOccurs="0" />
```

## 2.293.23 Style.Numerallanguage

The **Style.Numerallanguage** element specifies the digit format to use as described by a language. This element is optional.

If the **Style.Numerallanguage** element is present, its value MUST be a [ReportLanguage](#). If the **Style.Numerallanguage** element is not present, its value MUST be the same as that of the peer [Style.Language](#) element.

For text-formatting operations, the **Style.Numerallanguage** element MUST apply only to the following:

- [Textbox.Value](#)
- [DataLabel.Value](#)
- [ChartMember.Label](#)
- [TextRun.Value](#)
- [DataValue.Value](#)

If the grandparent element of the **Style.Numerallanguage** element is not [TextRun](#), [Textbox](#), [Chart](#), [LinearGauge](#), or [RadialGauge](#), the **Style.Numerallanguage** element is ignored.

Following is the parent element of the **Style.NumeralLanguage** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.NumeralLanguage** element.

```
<xsd:element name="NumeralLanguage" type="xsd:string" minOccurs="0" />
```

## 2.293.24 Style.NumeralVariant

The **Style.NumeralVariant** element specifies the **variant** of the digit format to use for text within a report item. This element is optional.

If this element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2](#) section 3.3.17) or an expression that evaluates to an **Integer**. The value of the **Style.NumeralVariant** element MUST be greater than or equal to 1 and less than or equal to 7. The values of this element are defined as follows:

- 1: Default (follow Unicode context rules)
- 2: 0123456789
- 3: Traditional digits for the script as defined in GDI+. Applicable only to numeral languages with the following cultures: ar, bn, bo, fa, gu, hi, kn, kok, lo, mr, ms, or, pa, sa, ta, te, th, ur
- 4: Applicable only to numeral languages with the following cultures: ko, ja, zh-CHS, zh-CHT
- 5: Applicable only to numeral languages with the following cultures: ko, ja, zh-CHS, zh-CHT
- 6: Applicable only to numeral languages with the following cultures: ko, ja, zh-CHS, zh-CHT (wide versions of regular digits)
- 7: Applicable only to numeral languages with the following cultures: ko

If the **Style.NumeralVariant** element is not present, its value is interpreted as 1. If the grandparent element of the **Style.NumeralVariant** element is not [TextRun](#), [Textbox](#), [Chart](#), [LinearGauge](#), or [RadialGauge](#), the **Style.NumeralVariant** element is ignored.

Following is the parent element of the **Style.NumeralVariant** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.NumeralVariant** element.

```
<xsd:element name="NumeralVariant" type="xsd:string" minOccurs="0" />
```

## 2.293.25 Style.PaddingBottom

The **Style.PaddingBottom** element specifies the padding between the bottom edge of a report item and its contents. This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**.

The value of this element MUST NOT be a size that is greater than 1000 pt or the equivalent. If the grandparent element of the **Style.PaddingBottom** element is not [Textbox](#), [Subtotal](#), or [Image](#), the **Style.PaddingBottom** element is ignored.

Following is the parent element of the **Style.PaddingBottom** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.PaddingBottom** element.

```
<xsd:element name="PaddingBottom" type="xsd:string" minOccurs="0" />
```

### 2.293.26 **Style.PaddingLeft**

The **Style.PaddingLeft** element specifies the padding between the left edge of a report item and its contents. This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**.

The value of the **Style.PaddingLeft** element MUST NOT be a size that is greater than 1000 pt or the equivalent. If the grandparent element of this element is not [Textbox](#), [Subtotal](#), or [Image](#), the **Style.PaddingLeft** element is ignored.

Following is the parent element of the **Style.PaddingLeft** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.PaddingLeft** element.

```
<xsd:element name="PaddingLeft" type="xsd:string" minOccurs="0" />
```

### 2.293.27 **Style.PaddingRight**

The **Style.PaddingRight** element specifies the padding between the right edge of a report item and its contents. This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**.

The value of the **Style.PaddingRight** element MUST NOT be a size that is greater than 1000 pt or the equivalent. If the grandparent element of this element is not [Textbox](#), [Subtotal](#), or [Image](#), the **Style.PaddingRight** element is ignored.

Following is the parent element of the **Style.PaddingRight** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.PaddingRight** element.

```
<xsd:element name="PaddingRight" type="xsd:string" minOccurs="0" />
```

## 2.293.28 **Style.PaddingTop**

The **Style.PaddingTop** element specifies the padding between the top edge of a report item and its contents. This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**.

The value of the **Style.PaddingTop** element MUST NOT be a size that is greater than 1000 pt or the equivalent. If the grandparent element of the **Style.PaddingTop** element is not [Textbox](#), [Subtotal](#), or [Image](#), the **Style.PaddingTop** element is ignored.

Following is the parent element of the **Style.PaddingTop** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.PaddingTop** element.

```
<xsd:element name="PaddingTop" type="xsd:string" minOccurs="0" />
```

## 2.293.29 **Style.RightBorder**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Style.RightBorder** element specifies properties for the right border of a report item. These properties override the corresponding value in the default sibling [Style.Border](#) element. The **Style.RightBorder** element is optional and is of type [Border](#).

If the grandparent element of the **Style.RightBorder** element is not [Line](#), [Rectangle](#), [Textbox](#), [Image](#), [SubReport](#), [PageSection](#), [Tablix](#), [Chart](#), [GaugePanel](#), or [Page](#), the **Style.RightBorder** element is ignored.

Following is the parent element of the **Style.RightBorder** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.RightBorder** element.

```
<xsd:element name="RightBorder" type="BorderType" minOccurs="0" />
```

## 2.293.30 **Style.ShadowColor**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Style.ShadowColor** element specifies the color of a shadow for a report item. This element is optional. If this element is present, its element MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

If the **Style.ShadowColor** element is not present, its value is interpreted as "#00000007F". If the grandparent element of the **Style.ShadowColor** element is not [ChartArea](#), [PlotArea](#), [ChartTitle](#), [ChartLegend](#), or [ChartSeries](#), the **Style.ShadowColor** element is ignored.

Following is the parent element of the **Style.ShadowColor** element.

<b>Parent elements</b>
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.ShadowColor** element.

```
<xsd:element name="ShadowColor" type="xsd:string" minOccurs="0" />
```

### 2.293.31 **Style.ShadowOffset**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Style.ShadowOffset** element specifies the size of the shadow for a report item. This element is optional. If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**.

If the grandparent element of the **Style.ShadowOffset** element is not [ChartArea](#), [ChartTitle](#), [ChartLegend](#), [ChartSeries](#), [LinearPointer](#), [RadialPointer](#), [LinearScale](#), [RadialScale](#), or [ScaleRange](#), the **Style.ShadowOffset** element is ignored.

Following is the parent element of the **Style.ShadowOffset** element.

<b>Parent elements</b>
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.ShadowOffset** element.

```
<xsd:element name="ShadowOffset" type="xsd:string" minOccurs="0" />
```

### 2.293.32 **Style.TextAlign**

The **Style.TextAlign** element specifies the horizontal alignment for the text within a report item. This element is optional. If the **Style.TextAlign** element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies the default horizontal text alignment, which is the same as "General".

**General:** Specifies that the text alignment is derived from the data type and the peer [Style.Direction](#) element.

**Left:** Specifies that the text is left-aligned.

**Center:** Specifies that the text is center-aligned.

**Right:** Specifies that the text is right-aligned.

If the **Style.TextAlign** element is not present, its value is interpreted as "Default". If the grandparent element of the **Style.TextAlign** element is not [Paragraph](#), [Textbox](#), [Subtotal](#), [ChartTitle](#), [ChartLegend](#), [ChartLegendTitle](#), [ChartAxis](#), [Axis](#), [ChartAxisTitle](#), [Title](#), [ChartDataLabel](#), or [ChartStripLine](#), the **Style.TextAlign** element is ignored.

Following is the parent element of the **Style.TextAlign** element.

<b>Parent elements</b>
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.TextAlign** element.

```
<xsd:element name="TextAlign" type="xsd:string" minOccurs="0" />
```

### 2.293.33 **Style.TextDecoration**

The **Style.TextDecoration** element specifies any special text formatting for text in a report item. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies the default text formatting, which is the same as "None".

**None:** Specifies that there is no text formatting.

**Underline:** Specifies that the text is underlined.

**Overline:** Specifies that the text has an overline.

**LineThrough:** Specifies that the text has a strikethrough.

If the **Style.TextDecoration** element is not present, its value is interpreted as "Default". If the grandparent element of the **Style.TextDecoration** element is not [TextRun](#), [Textbox](#), [Subtotal](#), [ChartTitle](#), [ChartLegend](#), [ChartLegendTitle](#), [ChartAxis](#), [Axis](#), [ChartAxisTitle](#), [Title](#), [ChartDataLabel](#), or [ChartStripLine](#), the **Style.TextDecoration** element is ignored.

Following is the parent element of the **Style.TextDecoration** element.

<b>Parent elements</b>
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.TextDecoration** element.

```
<xsd:element name="TextDecoration" type="xsd:string" minOccurs="0" />
```

### 2.293.34 **Style.TextEffect**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Style.TextEffect** element specifies an effect to apply to the text within a report item. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies the default text effect, which is the same as "None".

**None:** Specifies that no special effect is applied to the text.

**Shadow:** Specifies that the text has a shadow.

**Emboss:** Specifies that the text is embossed.



**Embed:** Specifies that the text has an embedded effect.

**Frame:** Specifies that a frame is added around the text.

If the **Style.TextEffect** element is not present, its value is interpreted as "Default". If the grandparent element of this element is not [ChartTitle](#), the **Style.TextEffect** element is ignored.

Following is the parent element of the **Style.TextEffect** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.TextEffect** element.

```
<xsd:element name="TextEffect" type="xsd:string" minOccurs="0" />
```

## 2.293.35 **Style.TopBorder**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Style.TopBorder** element specifies properties for the top border of a report item. These properties override the corresponding value in the default sibling element [Style.Border](#). The **Style.TopBorder** element is optional and is of type [Border](#).

If the grandparent element of the **Style.TopBorder** element is not [Line](#), [Rectangle](#), [Textbox](#), [Image](#), [SubReport](#), [PageSection](#), [Tablix](#), [Chart](#), [GaugePanel](#), or [Page](#), the **Style.TopBorder** element is ignored.

Following is the parent element of the **Style.TopBorder** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.TopBorder** element.

```
<xsd:element name="TopBorder" type="BorderType" minOccurs="0" />
```

## 2.293.36 **Style.UnicodeBiDi**

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **Style.UnicodeBiDi** element specifies the level of embedding with respect to the bi-directional algorithm. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or expression that evaluates to one of the following:

**Normal:** Specifies that an additional level of embedding is not opened.

**Embed:** Specifies that an additional level of embedding is opened.

**BiDi-Override:** Specifies that an additional level of embedding is opened and that reordering is sequenced according to the direction. This value overrides the implicit bidirectional algorithm.

If the **Style.UnicodeBiDi** element is not present, its value is interpreted as "Normal". If the grandparent element of this element is not [Textbox](#) or [Subtotal](#), the **Style.UnicodeBiDi** element is ignored.

Following is the parent element of the **Style.UnicodeBiDi** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.UnicodeBiDi** element.

```
<xsd:element name="UnicodeBiDi" type="xsd:string" minOccurs="0" />
```

### 2.293.37 **Style.VerticalAlign**

The **Style.VerticalAlign** element specifies the vertical alignment of text within a report item. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies the default vertical text alignment, which is the same as "Top".

**Top:** Specifies top-aligned text.

**Middle:** Specifies vertically-centered text alignment.

**Bottom:** Specifies bottom-aligned text.

If the **Style.VerticalAlign** element is not present, its value is interpreted as "Default". If the grandparent element of the **Style.VerticalAlign** element is not [Textbox](#), [Subtotal](#), [ChartTitle](#), [ChartLegendTitle](#), [ChartAxis](#), [Axis](#), [ChartLegend](#), [ChartAxisTitle](#), [Title](#), [ChartDataLabel](#), or [ChartStripLine](#), the **Style.VerticalAlign** element is ignored.

Following is the parent element for the **Style.VerticalAlign** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.VerticalAlign** element.

```
<xsd:element name="VerticalAlign" type="xsd:string" minOccurs="0" />
```

### 2.293.38 **Style.WritingMode**

The **Style.WritingMode** element specifies whether text is written vertically or horizontally. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. In [RDL 2003/10](#), [RDL 2005/01](#), and [RDL 2008/01](#), the value of this element MUST be one of the following or an expression that evaluates to one of the following. In [RDL 2010/01](#) and [RDL 2016/01](#), the value of this element MUST be one of the following.

**Default:** Specifies the default mode of writing text, which is the same as "Horizontal".

**Horizontal:** Specifies horizontal text.

**Vertical:** Specifies vertical text. This text rotates individual text characters (except East Asian text characters) 90 degrees.

If the **Style.WritingMode** element is not present, its value is interpreted as "Default". If the grandparent element of the **Style.WritingMode** element is not [Textbox](#) or [Subtotal](#), the **Style.WritingMode** element is ignored.

Following is the parent element of the **Style.WritingMode** element.

Parent elements
<a href="#">Style</a>

The following is the XML Schema definition of the **Style.WritingMode** element.

```
<xsd:element name="WritingMode" type="xsd:string" minOccurs="0" />
```

## 2.294 BackgroundImage

The **BackgroundImage** element specifies the **background images** for a report item. If the grandparent element of the **BackgroundImage** element is not [Rectangle](#), [Textbox](#), [Tablix](#), [Body](#), [PageSection](#), [PageHeaderFooter](#), [Page](#), [List](#), [Table](#), [Matrix](#), [Subtotal](#), [Chart](#), [ChartArea](#), [ChartDataPoint](#), [ChartEmptyPoints](#), [ChartMarker](#), [Marker](#), or [ChartStripLine](#), the **BackgroundImage** element is ignored.

The following are the parent and child elements of the **BackgroundImage** element.

Parent elements
<a href="#">Style</a>

Child elements
<a href="#">BackgroundImage.BackgroundRepeat</a>
<a href="#">BackgroundImage.MIMETYPE</a>
<a href="#">BackgroundImage.Position</a>
<a href="#">BackgroundImage.Source</a>
<a href="#">BackgroundImage.TransparentColor</a>
<a href="#">BackgroundImage.Value</a>

**Applies to** [RDL 2012/01](#)

Child elements
<a href="#">BackgroundImage.EmbeddingMode</a>
<a href="#">BackgroundImage.Transparency</a>

The following is the XML Schema definition of the **BackgroundImage** element.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections [5.5](#), [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:complexType name="BackgroundImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Source">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="External" />
          <xsd:enumeration value="Embedded" />
          <xsd:enumeration value="Database" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundRepeat" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.294.1 BackgroundImage.BackgroundRepeat

The **BackgroundImage.BackgroundRepeat** element specifies how a [BackgroundImage](#) fills the available space within its container. The **BackgroundImage.BackgroundRepeat** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies the default **BackgroundImage** repeating behavior, which is the same as "Fit" in a [Chart](#) element and "Repeat" elsewhere ([RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)).

**Repeat:** Specifies that the image repeats both horizontally and vertically to fill the space.

**RepeatX:** Specifies that the image repeats horizontally to fill the space. This value MUST NOT be allowed if the containing report item is a chart.

**RepeatY:** Specifies that the image repeats vertically to fill the space. This value MUST NOT be allowed if the containing report item is a chart.

**NoRepeat:** Specifies that the image does not repeat to fill the space ([RDL 2003/10](#) and [RDL 2005/01](#)).

**Fit:** Specifies that the image stretches to fill the space. This value MUST be allowed only when the containing report item is a chart ([RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)) or only when **Page** is the grandparent element of **BackgroundImage** ([RDL 2012/01](#)).

**Clip:** Specifies that the image is clipped to the available space ([RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)).

**FitProportional:** Specifies that the image stretches to fill the space and, at the same time, preserve its aspect ratio. This value MUST be allowed only when **Page** is the grandparent element of **BackgroundImage** ([RDL 2012/01](#)).

If the **BackgroundImage.BackgroundRepeat** element is not present, its value is interpreted as "Default" in RDL 2008/01, RDL 2010/01, and RDL 2016/01, and its value is interpreted as "Repeat" in RDL 2003/10 and RDL 2005/01. If the parent **BackgroundImage** element is not ignored but the grandparent element of the **BackgroundImage** element is [ChartDataPoint](#), [ChartEmptyPoints](#), [Marker](#), or [ChartMarker](#), the **BackgroundImage.BackgroundRepeat** element is ignored.

The following is the parent element of the **BackgroundImage.BackgroundRepeat** element.

Parent elements
BackgroundImage

The following is the XML Schema definition of the **BackgroundImage.BackgroundRepeat** element.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections [5.5](#), [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:element name="BackgroundRepeat" type="xsd:string" minOccurs="0" />
```

### 2.294.2 BackgroundImage.MIMETYPE

The **BackgroundImage.MIMETYPE** element specifies the image format of a [BackgroundImage](#). The **BackgroundImage.MIMETYPE** element is optional. If this element is present, its value MUST be a [ReportMIMETYPE](#).

If the peer [BackgroundImage.Source](#) element is set to a value other than "Database", the **BackgroundImage.MIMETYPE** element is ignored.

Following is the parent element of the **BackgroundImage.MIMETYPE** element.

Parent elements
BackgroundImage

The following is the XML Schema definition of the **BackgroundImage.MIMETYPE** element.

```
<xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
```

### 2.294.3 BackgroundImage.Position

The **BackgroundImage.Position** element specifies where a [BackgroundImage](#) element that has the value of the [BackgroundImage.BackgroundRepeat](#) element set to "Clip" is drawn. The **BackgroundImage.Position** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

The value of the **BackgroundImage.Position** element MUST be one of the following:

**Default:** Specifies the default position of the **background images**, which is the same as "Center" for charts. In [RDL 2012/01](#), Default is interpreted as "TopLeft" for pages.

**Top:** Specifies that the background image is drawn at the top center of the containing report item.

**TopLeft:** Specifies that the background image is drawn at the top left of the containing report item.

**TopRight:** Specifies that the background image is drawn at the top right of the containing report item.

**Left:** Specifies that the background image is drawn at center left of the containing report item.

**Center:** Specifies that the background image is drawn at the center of the containing report item.

**Right:** Specifies that the background image is drawn at the center right of the containing report item.

**BottomRight:** Specifies that the background image is drawn at the bottom right of the containing report item.

**Bottom:** Specifies that the background image is drawn at the bottom center of the containing report item.

**BottomLeft:** Specifies that the background image is drawn at the bottom left of the containing report item.

If the **BackgroundImage.Position** element is not present, its value is interpreted as "Default". If the parent **BackgroundImage** element is not ignored but the grandparent element of the **BackgroundImage** is [ChartDataPoint](#), [ChartEmptyPoints](#), [Marker](#), or [ChartMarker](#), the **BackgroundImage.Position** element is ignored.

If the peer **BackgroundImage.BackgroundRepeat** element is not specified as "Clip", the **BackgroundImage.Position** element is ignored.

Following is the parent element of the **BackgroundImage.Position** element.

Parent elements
BackgroundImage

The following is the XML Schema definition of the **BackgroundImage.Position** element.

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections [5.5](#), [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
```

#### 2.294.4 BackgroundImage.Source

The **BackgroundImage.Source** element specifies the type of source that is associated with a [BackgroundImage](#). The **BackgroundImage.Source** element MUST be specified. The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1), and it MUST be one of the following:

**External:** Specifies that the peer [BackgroundImage.Value](#) element contains a **String** constant or an expression that evaluates to the location of an image.

**Embedded:** Specifies that the peer **BackgroundImage.Value** element contains a **String** constant or an expression that evaluates to the name of an [EmbeddedImage](#) within a report.

**Database:** Specifies that the peer **BackgroundImage.Value** element contains an expression (such as a field in the database) that evaluates to the binary data for an image.

Following is the parent element of the **BackgroundImage.Source** element.

<b>Parent elements</b>
------------------------

BackgroundImage
-----------------

The following is the XML Schema definition of the **BackgroundImage.Source** element.

```
<xsd:element name="Source">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="External" />
      <xsd:enumeration value="Embedded" />
      <xsd:enumeration value="Database" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.294.5 BackgroundImage.TransparentColor

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **BackgroundImage.TransparentColor** element specifies the color to treat as transparent in a [BackgroundImage](#). The **BackgroundImage.TransparentColor** element is optional. If this element is present, its value MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

If the parent **BackgroundImage** element is not ignored but the grandparent of the **BackgroundImage** is [ChartDataPoint](#) or [ChartEmptyPoints](#), the **BackgroundImage.TransparentColor** element is ignored.

Following is the parent element of the **BackgroundImage.TransparentColor** element.

<b>Parent elements</b>
------------------------

BackgroundImage
-----------------

The following is the XML Schema definition of the **BackgroundImage.TransparentColor** element.

```
<xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
```

## 2.294.6 BackgroundImage.Value

The **BackgroundImage.Value** element specifies either the location or the actual data of a [BackgroundImage](#), depending on the value of the peer [BackgroundImage.Source](#) element. The **BackgroundImage.Value** element MUST be specified.

If the peer **BackgroundImage.Source** element is set to "External", the value of the **BackgroundImage.Value** element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to the location of an image. This location MUST be a [ReportPath](#) or [RdlURL](#) value.

If the peer **BackgroundImage.Source** element is set to "Embedded", the value of the **BackgroundImage.Value** element MUST be a **String** or an expression that evaluates to the name of an [EmbeddedImage](#) in a report.

If the peer **BackgroundImage.Source** element is set to "Database", the value of the **BackgroundImage.Value** element MUST be a **String** or an expression that evaluates to the binary data for an image.

If the **BackgroundImage.Value** element has an empty value, an image MUST NOT be displayed.

Following is the parent element of the **BackgroundImage.Value** element.

Parent elements
<a href="#">BackgroundImage</a>

The following is the XML Schema definition of the **BackgroundImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" />
```

## 2.294.7 BackgroundImage.EmbeddingMode

**Applies to** [RDL 2012/01](#)

The **BackgroundImage.EmbeddingMode** element specifies how the report embeds the **background image**. The **BackgroundImage.EmbeddingMode** element is optional. The **BackgroundImage.EmbeddingMode** element MUST NOT be specified unless the peer [BackgroundImage.Source](#) property is set to "Embedded". The value of the **BackgroundImage.EmbeddingMode** element MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) that is one of the following:

**Inline:** Specifies that the value of the peer [BackgroundImage.Value](#) element is a **String** or an expression that evaluates to the name of an [EmbeddedImage](#) in a report.

**Package:** Specifies that the report package includes the image as a package part with the appropriate image content type. The value contains a constant or an expression that evaluates to the **ID** of the package relationship (that is, "rld1"). For more details, see [\[MS-DPRDL\]](#) section 2.2.1 and [\[ECMA-376-2/2\]](#).

Following is the parent element of the **BackgroundImage.EmbeddingMode** element.

Parent elements
<a href="#">BackgroundImage</a>

The following is the XML Schema definition of the **BackgroundImage.EmbeddingMode** element.

```
<xsd:element name="EmbeddingMode">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Inline"/>
      <xsd:enumeration value="Package"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.294.8 BackgroundImage.Transparency

**Applies to** [RDL 2012/01](#)

The **BackgroundImage.Transparency** element specifies the percent transparency for the **background image**. The **BackgroundImage.Transparency** element is optional. The range of valid values is from 0 (completely opaque) to 100 (completely transparent).

Following is the parent element of the **BackgroundImage.Transparency** element.



<b>Parent elements</b>
------------------------

<a href="#">BackgroundImage</a>
---------------------------------

The following is the XML Schema definition of the **BackgroundImage.Transparency** element.

```
<xsd:element name="Transparency" type="xsd:string" />
```

## 2.295 Border

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Border** element specifies the default border properties for a report item. This element is ignored if it applies to [Paragraph](#), [TextRun](#), [ChartAxisTitle](#), [Title](#), [LinearGauge](#), [RadialGauge](#), [Thermometer](#), [PointerCap](#), [ScaleLabels](#), [PinLabel](#), or [FrameBackground](#).

The following are the parent and child elements of the **Border** element.

<b>Parent elements</b>
------------------------

<a href="#">Style</a>
-----------------------

<b>Child elements</b>
-----------------------

<a href="#">Border.Color</a>
------------------------------

<a href="#">Border.Style</a>
------------------------------

<a href="#">Border.Width</a>
------------------------------

The following is the XML Schema definition of the **Border** element in RDL 2008/01.

```
<xsd:complexType name="BorderType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Color" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **Border** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="BorderType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Color" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

## 2.295.1 Border.Color

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Border.Color** element specifies the color of a [Border](#) element. The **Border.Color** element is optional. The value of the **Border.Color** element MUST be an [RdlColor](#) or an expression that evaluates to an **RdlColor**.

Following is the parent element of the **Border.Color** element.

Parent elements
Border

The following is the XML Schema definition of the **Border.Color** element.

```
<xsd:element name="Color" type="xsd:string" minOccurs="0" />
```

## 2.295.2 Border.Style

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Border.Style** element specifies the style of a [Border](#) element. The **Border.Style** element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. The value of this element MUST be one of the following or an expression that evaluates to one of the following:

**Default:** Specifies the default style for a **Border** element, which is the same as "Solid" if the **Border** element is in a [Line](#) element and which is "None" otherwise.

**None:** Specifies that no border is drawn.

**Dotted:** Specifies a dotted line for a border.

**Dashed:** Specifies a dashed line for a border.

**Solid:** Specifies a solid line for a border.

**Double:** Specifies a double solid line for a border.

**DashDot:** Specifies a dash-dot pattern line for a border. This value is permitted only within a [Chart](#) element.

**DashDotDot:** Specifies a dash-dot-dot pattern line for a border. This value is permitted only within a **Chart** element.

If the **Border.Style** element is not present, its value is interpreted as "Default".

Following is the parent element of the **Border.Style** element.

Parent elements
Border

The following is the XML Schema definition of the **Border.Style** element.

```
<xsd:element name="Style" type="xsd:string" minOccurs="0" />
```

### 2.295.3 Border.Width

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Border.Width** element specifies the width of the [Border](#) of a report item. The **Border.Width** element is optional. If this element is present, its value **MUST** be [RdlSize](#) or an expression that evaluates to an **RdlSize**.

If the **Border.Width** element is not present, its value is interpreted as 1 pt or the equivalent. The value of this element **MUST NOT** be a size that is greater than 20 pt or the equivalent. The value of the **Border.Width** element **MUST NOT** be a size that is less than 0.25 pt or the equivalent.

Following is the parent element of the **Border.Width** element.

Parent elements
<a href="#">Border</a>

The following is the XML Schema definition of the **Border.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
```

### 2.296 BorderColor

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderColor** element specifies the color properties of the [Border](#) of a report item.

The following are the parent and child elements of the **BorderColor** element.

Parent elements
<a href="#">Style</a>

Child elements
<a href="#">BorderColor.Bottom</a>
<a href="#">BorderColor.Default</a>
<a href="#">BorderColor.Left</a>
<a href="#">BorderColor.Right</a>
<a href="#">BorderColor.Top</a>

The following is the XML Schema definition of the **BorderColor** element.

```
<xsd:complexType name="BorderColorStyleWidthType">  
  <xsd:choice minOccurs="0" maxOccurs="unbounded">  
    <xsd:element name="Default" type="xsd:string" minOccurs="0" />  
  </xsd:choice>  
</xsd:complexType>
```

```

<xsd:element name="Left" type="xsd:string" minOccurs="0" />
<xsd:element name="Right" type="xsd:string" minOccurs="0" />
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
<xsd:element name="Bottom" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.296.1 **BorderColor.Bottom**

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderColor.Bottom** element specifies the color of a bottom border. This element is optional. If this element is present, its value MUST be an [RdlColor](#). If this element is not present, its value is interpreted to be the same value as that of [BorderColor.Default](#).

Following is the parent element of the **BorderColor.Bottom** element.

Parent elements
<a href="#">BorderColor</a>

The following is the XML Schema definition of the **BorderColor.Bottom** element.

```
<xsd:element name="Bottom" type="xsd:string" minOccurs="0" />
```

### 2.296.2 **BorderColor.Default**

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderColor.Default** element specifies the color of all borders for a report item (unless the color is overridden for a specific side). This element is optional. If this element is present, its value MUST be an [RdlColor](#). If this element is not present, its value is interpreted as "Black".

Following is the parent element of the **BorderColor.Default** element.

Parent elements
<a href="#">BorderColor</a>

The following is the XML Schema definition of the **BorderColor.Default** element.

```
<xsd:element name="Default" type="xsd:string" minOccurs="0" />
```

### 2.296.3 **BorderColor.Left**

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderColor.Left** element specifies the color of the left [Border](#) of a report item. This element is optional. If this element is present, its value MUST be an [RdlColor](#). If this element is not present, its value is interpreted to be the same value as that of [BorderColor.Default](#).

Following is the parent element of the **BorderColor.Left** element.

<b>Parent elements</b>
------------------------

<a href="#">BorderColor</a>
-----------------------------

The following is the XML Schema definition of the **BorderColor.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

## 2.296.4 BorderColor.Right

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **BorderColor.Right** element specifies the color of the right [Border](#) of a report item. This element is optional. If this element is present, its value MUST be an [RdlColor](#). If this element is not present, its value is interpreted to be the same value as that of [BorderColor.Default](#).

Following is the parent element of the **BorderColor.Right** element.

<b>Parent elements</b>
------------------------

<a href="#">BorderColor</a>
-----------------------------

The following is the XML Schema definition of the **BorderColor.Right** element.

```
<xsd:element name="Right" type="xsd:string" minOccurs="0" />
```

## 2.296.5 BorderColor.Top

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

The **BorderColor.Top** element specifies the color of the top [Border](#) of a report item. This element is optional. If this element is present, its value MUST be an [RdlColor](#). If this element is not present, its value is interpreted as the same value as that of [BorderColor.Default](#).

Following is the parent element of the **BorderColor.Top** element.

<b>Parent elements</b>
------------------------

<a href="#">BorderColor</a>
-----------------------------

The following is the XML Schema definition of the **BorderColor.Top** element.

```
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
```

## 2.297 BorderStyle

*Applies to [RDL 2003/10](#), [RDL 2005/01](#), and [RDL 2008/01](#)*

The **BorderStyle** element specifies appearance properties for a [Border](#) for a report item.

The following are the parent and child elements of the **BorderStyle** element.

<b>Parent elements</b>
<a href="#">Style</a>

<b>Child elements</b>
<a href="#">BorderStyle.Bottom</a>
<a href="#">BorderStyle.Default</a>
<a href="#">BorderStyle.Left</a>
<a href="#">BorderStyle.Right</a>
<a href="#">BorderStyle.Top</a>

The following is the XML Schema definition of the **BorderStyle** element.

```
<xsd:complexType name="BorderColorStyleWidthType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Default" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Right" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Bottom" type="xsd:string" minOccurs="0" />
  <xsd:any namespace="#" />
</xsd:complexType>
```

### 2.297.1 **BorderStyle.Bottom**

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderStyle.Bottom** element specifies the style of the top border. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to one of the following values:

**None:** Specifies that no border is drawn.

**Dotted:** Specifies a dotted line for a border.

**Dashed:** Specifies a dashed line for a border.

**Solid:** Specifies a solid line for a border.

**Double:** Specifies a double solid line for a border.

**Groove:** Specifies a groove line for a border.

**Ridge:** Specifies a ridge line for a border.

**Inset:** Specifies an inset line for a border.

**WindowInset:** Specifies a window inset line for a border.

**Outset:** Specifies an outset line for a border.

If this element is not present, its value is interpreted as "None".

Following is the parent element of the **BorderStyle.Bottom** element.

<b>Parent elements</b>
------------------------

<a href="#">BorderStyle</a>
-----------------------------

The following is the XML Schema definition of the **BorderStyle.Bottom** element.

```
<xsd:element name="Bottom" type="xsd:string" minOccurs="0" />
```

## 2.297.2 BorderStyle.Default

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderStyle.Default** element specifies the style for all borders of a report item (unless the style is overridden for a specific side). This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to one of the following values:

**None:** Specifies that no border is drawn.

**Dotted:** Specifies a dotted line for a border.

**Dashed:** Specifies a dashed line for a border.

**Solid:** Specifies a solid line for a border.

**Double:** Specifies a double solid line for a border.

**Groove:** Specifies a groove line for a border.

**Ridge:** Specifies a ridge line for a border.

**Inset:** Specifies an inset line for a border.

**WindowInset:** Specifies a window inset line for a border.

**Outset:** Specifies an outset line for a border.

If the **BorderStyle.Default** element is not present, its value is interpreted as "None". When the **BorderStyle.Default** element is used in a [Chart](#), a **plot area**, or a legend, its value MUST NOT be set to "Double", "Groove", "Ridge", "Inset", "WindowInset", or "Outset".

Following is the parent element of the **BorderStyle.Default** element.

<b>Parent elements</b>
------------------------

<a href="#">BorderStyle</a>
-----------------------------

The following is the XML Schema definition of the **BorderStyle.Default** element.

```
<xsd:element name="Default" type="xsd:string" minOccurs="0" />
```

## 2.297.3 BorderStyle.Left

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderStyle.Left** element specifies the style of a left border. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to one of the following values:

**None:** Specifies that no border is drawn.

**Dotted:** Specifies a dotted line for a border.

**Dashed:** Specifies a dashed line for a border.

**Solid:** Specifies a solid line for a border.

**Double:** Specifies a double solid line for a border.

**Groove:** Specifies a groove line for a border.

**Ridge:** Specifies a ridge line for a border.

**Inset:** Specifies an inset line for a border.

**WindowInset:** Specifies a window inset line for a border.

**Outset:** Specifies an outset line for a border.

If the **BorderStyle.Left** element is not present, its value is interpreted as "None".

Following is the parent element of the **BorderStyle.Left** element.

Parent elements
<a href="#">BorderStyle</a>

The following is the XML Schema definition of the **BorderStyle.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

## 2.297.4 **BorderStyle.Right**

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderStyle.Right** element specifies the style of a right border. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to one of the following values:

**None:** Specifies that no border is drawn.

**Dotted:** Specifies a dotted line for a border.

**Dashed:** Specifies a dashed line for a border.

**Solid:** Specifies a solid line for a border.

**Double:** Specifies a double solid line for a border.

**Groove:** Specifies a groove line for a border.

**Ridge:** Specifies a ridge line for a border.

**Inset:** Specifies an inset line for a border.



**WindowInset:** Specifies a window inset line for a border.

**Outset:** Specifies an outset line for a border.

If the **BorderStyle.Right** element is not present, its value is interpreted as "None".

Following is the parent element of the **BorderStyle.Right** element.

Parent elements
<a href="#">BorderStyle</a>

The following is the XML Schema definition of the **BorderStyle.Right** element.

```
<xsd:element name="Right" type="xsd:string" minOccurs="0" />
```

## 2.297.5 BorderStyle.Top

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderStyle.Top** element specifies the style of a top border. This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to one of the following values:

**None:** Specifies that no border is drawn.

**Dotted:** Specifies a dotted line for a border.

**Dashed:** Specifies a dashed line for a border.

**Solid:** Specifies a solid line for a border.

**Double:** Specifies a double solid line for a border.

**Groove:** Specifies a groove line for a border.

**Ridge:** Specifies a ridge line for a border.

**Inset:** Specifies an inset line for a border.

**WindowInset:** Specifies a window inset line for a border.

**Outset:** Specifies an outset line for a border.

If the **BorderStyle.Top** element is not present, its value is interpreted as "None".

Following is the parent element of the **BorderStyle.Top** element.

Parent elements
<a href="#">BorderStyle</a>

The following is the XML Schema definition of the **BorderStyle.Top** element.

```
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
```

## 2.298 BorderWidth

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderWidth** element specifies width properties for a border.

The following are the parent and child elements of the **BorderWidth** element.

Parent elements
<a href="#">Style</a>

Child elements
<a href="#">BorderWidth.Bottom</a>
<a href="#">BorderWidth.Default</a>
<a href="#">BorderWidth.Left</a>
<a href="#">BorderWidth.Right</a>
<a href="#">BorderWidth.Top</a>

The following is the XML Schema definition of the **BorderWidth** element.

```
<xsd:complexType name="BorderColorStyleWidthType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Default" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Right" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Bottom" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.298.1 BorderWidth.Bottom

Applies to [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderWidth.Bottom** element specifies the width of the bottom [Border](#) of a report item. This element is optional. If this element is present, its value MUST be an [RdlSize](#) or expression that evaluates to an **RdlSize**. The value of the **BorderWidth.Bottom** element MUST NOT be less than 0.25 pt or the equivalent, and it MUST NOT exceed 20 pt or the equivalent.

Following is the parent element of the **BorderWidth.Bottom** element.

Parent elements
<a href="#">BorderWidth</a>

The following is the XML Schema definition of the **BorderWidth.Bottom** element.

```
<xsd:element name="Bottom" type="xsd:string" minOccurs="0" />
```

## 2.298.2 BorderWidth.Default

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderWidth.Default** element specifies the width of all borders of a report item (unless the width is overridden for a specific side). Borders SHOULD be centered on the edge of the report item. This element is optional.

If this element is present, its value MUST be an [RdlSize](#) or an expression that evaluates to an **RdlSize**. The **RdlSize** value MUST NOT be less than 0.25 pt or the equivalent, and it MUST NOT exceed 20 pt or the equivalent.

If this element is not present, its value is interpreted as 1 pt.

Following is the parent element of the **BorderWidth.Default** element.

Parent elements
<a href="#">BorderWidth</a>

The following is the XML Schema definition of the **BorderWidth.Default** element.

```
<xsd:element name="Default" type="xsd:string" minOccurs="0" />
```

## 2.298.3 BorderWidth.Left

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderWidth.Left** element specifies the width of the left [Border](#) of a report item. This element is optional. If this element is present, its value MUST be an [RdlSize](#) or expression that evaluates to an **RdlSize**. The value of this element MUST NOT be less than 0.25 pt or the equivalent, and it MUST NOT exceed 20 pt or the equivalent.

Following is the parent element of the **BorderWidth.Left** element.

Parent elements
<a href="#">BorderWidth</a>

The following is the XML Schema definition of the **BorderWidth.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

## 2.298.4 BorderWidth.Right

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderWidth.Right** element specifies the width of the right [Border](#) of a report item. This element is optional. If this element is present, its value MUST be an [RdlSize](#) or expression that evaluates to an **RdlSize**. The value of this element MUST NOT be less than 0.25 pt or the equivalent, and it MUST NOT exceed 20 pt or the equivalent.

Following is the parent element of the **BorderWidth.Right** element.

<b>Parent elements</b>
<a href="#">BorderWidth</a>

The following is the XML Schema definition of the **BorderWidth.Right** element.

```
<xsd:element name="Right" type="xsd:string" minOccurs="0" />
```

## 2.298.5 BorderWidth.Top

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

The **BorderWidth.Top** element specifies the width of the top [Border](#) of a report item. This element is optional. If this element is present, its value MUST be an [RdSize](#) or expression that evaluates to an **RdSize**. The value of the **BorderWidth.Top** element MUST NOT be less than 0.25 pt or the equivalent, and it MUST NOT exceed 20 pt or the equivalent.

Following is the parent element of the **BorderWidth.Top** element.

<b>Parent elements</b>
<a href="#">BorderWidth</a>

The following is the XML Schema definition of the **BorderWidth.Top** element.

```
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
```

## 2.299 Parameters

The **Parameters** element specifies the ordered list of [Parameter](#) elements for [Subreport.Parameters](#) or [Drillthrough.Parameters](#). This list MUST contain at least one parameter.

The following are the parent and child elements of the **Parameters** element.

<b>Parent elements</b>
<a href="#">Subreport</a>
<a href="#">Drillthrough</a>

<b>Child elements</b>
<a href="#">Parameters.Parameter</a>

The following is the XML Schema definition of the **Parameters** element.

```
<xsd:complexType name="ParametersType">
  <xsd:sequence>
    <xsd:element name="Parameter" type="ParameterType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

## 2.299.1 Parameters.Parameter

The **Parameters.Parameter** element specifies a [Parameter](#) in the collection of [Parameters](#) for the subreport or drillthrough. This element **MUST** be specified. At least one **Parameter** element **MUST** be specified for a **Parameters** collection, and NULL is a valid value for the [Parameter.Value](#) element. This element is of type **Parameter**.

Following is the parent element of the **Parameters.Parameter** element.

Parent elements
<a href="#">Parameters</a>

The following is the XML Schema definition of the **Parameters.Parameter** element.

```
<xsd:element name="Parameter" type="ParameterType" maxOccurs="unbounded" />
```

## 2.300 Parameter

The **Parameter** element specifies information about a subreport or **drillthrough parameter**.

The following are the attributes and child elements of the **Parameter** element.

Attributes
<a href="#">Parameter.Name</a>

Child elements
<a href="#">Parameter.Omit</a>
<a href="#">Parameter.Value</a>

The following is the XML Schema definition of the **Parameter** element.

```
<xsd:complexType name="ParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="Omit" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.300.1 Parameter.Name

The **Parameter.Name** attribute specifies the name of the [ReportParameter](#) of a dependent [Report](#). This attribute **MUST** be specified. The value of this attribute **MUST** be a **CLS-compliant identifier** [\[UTR15\]](#).

Following is the parent element of the **Parameter.Name** attribute.

<b>Parent elements</b>
------------------------

<a href="#">Parameter</a>
---------------------------

The following is the XML Schema definition of the **Parameter.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:string" use="required" />
```

### 2.300.2 Parameter.Omit

The **Parameter.Omit** element specifies that the parameter is not used. This element is ignored when it is used in a parameter within [Subreport.Parameters](#).

The **Parameter.Omit** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**. If the **Parameter.Omit** element is not present, its value is interpreted as false.

Following is the parent element of the **Parameter.Omit** element.

<b>Parent elements</b>
------------------------

<a href="#">Parameter</a>
---------------------------

The following is the XML Schema definition of the **Parameter.Omit** element.

```
<xsd:element name="Omit" type="xsd:string" minOccurs="0" />
```

### 2.300.3 Parameter.Value

The **Parameter.Value** element specifies the value or expression that evaluates to the value to pass in for the **report parameter** to the subreport or drillthrough. The **Parameter.Value** element MUST be specified, and its value MUST be a **VARIANT** or an expression that evaluates to a **VARIANT**.

Following is the parent element of the **Parameter.Value** element.

<b>Parent elements</b>
------------------------

<a href="#">Parameter</a>
---------------------------

The following is the XML Schema definition of the **Parameter.Value** element.

```
<xsd:element name="Value" type="xsd:string" />
```

## 2.301 Visibility

The **Visibility** element specifies properties to determine whether a report item is shown in a rendered report. This element is optional. If this element is not present, the report item MUST be shown unconditionally.

Following are the parent and child elements of the **Visibility** element.

Parent elements
<a href="#">Image</a>
<a href="#">Line</a>
<a href="#">Rectangle</a>
<a href="#">Subreport</a>
<a href="#">Textbox</a>
<a href="#">Tablix</a>
<a href="#">TablixMember</a>
<a href="#">List</a>
<a href="#">Matrix</a>
<a href="#">DynamicColumns</a>
<a href="#">DynamicRows</a>
<a href="#">Table</a>
<a href="#">Details</a>
<a href="#">TableRow</a>
<a href="#">TableColumn</a>
<a href="#">TableGroup</a>
<a href="#">Chart</a>
<a href="#">GaugePanel</a>
<a href="#">Map</a>
<a href="#">CustomReportItem</a>

Child elements
<a href="#">Visibility.Hidden</a>
<a href="#">Visibility.ToggleItem</a>

The following is the XML Schema definition of the **Visibility** element.

```
<xsd:complexType name="VisibilityType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToggleItem" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.301.1 Visibility.Hidden

The **Visibility.Hidden** element specifies whether a report item is initially hidden. This element is optional. If this element is not present, its value is interpreted as false. If the value of this element is true and the peer element [Visibility.ToggleItem](#) is not set, the report item that contains the **Visibility.Hidden** element MUST be hidden. The value of this element MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) or an expression that evaluates to a **Boolean**.

Following is the parent element of the **Visibility.Hidden** element.

Parent elements
<a href="#">Visibility</a>

The following is the XML Schema definition of the **Visibility.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0">
```

### 2.301.2 Visibility.ToggleItem

The **Visibility.ToggleItem** element specifies the name of a [Textbox](#) that is used to hide or unhide the containing report item. This element is optional.

If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1). The value of this element MUST contain the name of a text box in the same group as the containing report item or in any containing ancestor group scope. The value of this element MUST NOT refer to a report item within [PageSection](#) or [PageHeaderFooter](#) element. If the **Visibility.ToggleItem** element is not present, the appearance of the containing report item MUST NOT be toggleable.

Following is the parent element of the **Visibility.ToggleItem** element.

Parent elements
<a href="#">Visibility</a>

The following is the XML Schema definition of the **Visibility.ToggleItem** element.

```
<xsd:element name="ToggleItem" type="xsd:string" minOccurs="0">
```

## 2.302 Classes

The **Classes** element specifies information about the code **classes** to instantiate during the initialization of a [Report](#). These code class instances can then be used in expressions throughout a report. This element is optional.

The following are the parent and child elements of the **Classes** element.

Parent elements
Report



Child elements
----------------

<a href="#">Classes.Class</a>
-------------------------------

The following is the XML Schema definition of the **Classes** element.

```
<xsd:complexType name="ClassesType">
  <xsd:sequence>
    <xsd:element name="Class" type="ClassType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.302.1 Classes.Class

The **Classes.Class** element specifies individual **class** instances to be instantiated and used in a [Report](#). This element is of type [Class](#). This element MUST be specified at least once within a [Classes](#) collection.

Following is the parent element of the **Classes.Class** element.

Parent elements
-----------------

<a href="#">Classes</a>
-------------------------

The following is the XML Schema definition of the **Classes.Class** element.

```
<xsd:element name="Class" type="ClassType" maxOccurs="unbounded">
```

### 2.303 Class

The **Class** element specifies information about a **class** instance that is instantiated for a [Report](#) and that can be used in expressions or code in that report. This element MUST be specified at least once within a [Classes](#) collection. The **Class** element MUST contain both the [Class.ClassName](#) and [Class.InstanceName](#) elements.

The following are the parent and child elements of the **Class** element.

Parent elements
-----------------

<a href="#">Classes</a>
-------------------------

Child elements
----------------

<a href="#">Class.ClassName</a>
---------------------------------

<a href="#">Class.InstanceName</a>
------------------------------------

The following is the XML Schema definition of the **Class** element.

```
<xsd:complexType name="ClassType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ClassName" type="xsd:string" />
    <xsd:element name="InstanceName" type="xsd:normalizedString" />
  </xsd:choice>
</xsd:complexType>
```

```

    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.303.1 Class.ClassName

The **Class.ClassName** element specifies the name of a class to instantiate. The value of this element MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**. This element MUST be specified.

Following is the parent element of the **Class.ClassName** element.

Parent elements
<a href="#">Class</a>

The following is the XML Schema definition of the **Class.ClassName** element.

```
<xsd:element name="ClassName" type="xsd:string">
```

### 2.303.2 Class.InstanceName

The **Class.InstanceName** element specifies the name of an instance of a class to be instantiated and used in expressions throughout a [Report](#). This element MUST be specified. The value of the **Class.InstanceName** element MUST be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Class.InstanceName** element.

Parent elements
<a href="#">Class</a>

The following is the XML Schema definition of the **Class.InstanceName** element.

```
<xsd:element name="InstanceName" type="xsd:normalizedString">
```

## 2.304 CodeModules

The **CodeModules** element specifies the names of code modules to load and use within expressions in a [Report](#) or a [Report.Code](#) element. The **CodeModules** element is optional.

The following are the parent elements and the child element of the **CodeModules** element.

Parent elements
Report

<b>Child elements</b>
-----------------------

<a href="#">CodeModules.CodeModule</a>
--

The following is the XML Schema definition of the **CodeModules** element.

```
<xsd:complexType name="CodeModulesType">
  <xsd:sequence>
    <xsd:element name="CodeModule" type="xsd:string" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.304.1 CodeModules.CodeModule

The **CodeModules.CodeModule** element specifies the name of a code module to load. The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) or an expression that evaluates to a **String**. This element MUST be specified at least once within a [CodeModules](#) collection.

Following is the parent element of the **CodeModules.CodeModule** element.

<b>Parent elements</b>
------------------------

<a href="#">CodeModules</a>
-----------------------------

The following is the XML Schema definition of the **CodeModules.CodeModule** element.

```
<xsd:element name="CodeModule" type="xsd:string" maxOccurs="unbounded">
```

### 2.305 EmbeddedImages

The **EmbeddedImages** element specifies a collection of images that are embedded within a [Report](#). The **EmbeddedImages** element is optional.

The following are the parent and child elements of the **EmbeddedImages** element.

<b>Parent elements</b>
------------------------

<a href="#">Report</a>
------------------------

<b>Child elements</b>
-----------------------

<a href="#">EmbeddedImages.EmbeddedImage</a>
--

The following is the XML Schema definition of the **EmbeddedImages** element.

```
<xsd:complexType name="EmbeddedImagesType">
  <xsd:sequence>
    <xsd:element name="EmbeddedImage" type="EmbeddedImageType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.305.1 EmbeddedImages.EmbeddedImage

The **EmbeddedImages.EmbeddedImage** element specifies an image that is embedded within a [Report](#). The **EmbeddedImages.EmbeddedImage** element is of type [EmbeddedImage](#). This element MUST be specified at least once within an [EmbeddedImages](#) collection.

Following is the parent element of the **EmbeddedImages.EmbeddedImage** element.

Parent elements
<a href="#">EmbeddedImages</a>

The following is the XML Schema definition of the **EmbeddedImages.EmbeddedImage** element.

```
<xsd:element name="EmbeddedImage" type="EmbeddedImageType" maxOccurs="unbounded">
```

### 2.306 EmbeddedImage

The **EmbeddedImage** element specifies an image that is embedded within a [Report](#). The **EmbeddedImage** element MUST be specified at least once within an [EmbeddedImages](#) collection.

The following are the parent elements, attributes, and child elements of the **EmbeddedImage** element.

Parent elements
<a href="#">EmbeddedImages</a>

Attributes
<a href="#">EmbeddedImage.Name</a>

Child elements
<a href="#">EmbeddedImage.ImageData</a>
<a href="#">EmbeddedImage.MIMETYPE</a>

The following is the XML Schema definition of the **EmbeddedImage** element.

```
<xsd:complexType name="EmbeddedImageType">  
  <xsd:choice minOccurs="1" maxOccurs="unbounded">  
    <xsd:element name="MIMETYPE" type="xsd:string" />  
    <xsd:element name="ImageData" type="xsd:string" />  
    <xsd:any namespace="##other" processContents="skip" />  
  </xsd:choice>  
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

### 2.306.1 EmbeddedImage.Name

The **EmbeddedImage.Name** attribute specifies a unique identifier for an [EmbeddedImage](#). This attribute MUST be specified. The value of the **EmbeddedImage.Name** attribute MUST be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique within a [Report](#).

Following is the parent element of the **EmbeddedImage.Name** attribute.

Parent elements
EmbeddedImage

The following is the XML Schema definition of the **EmbeddedImage.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.306.2 EmbeddedImage.ImageData

The **EmbeddedImage.ImageData** element specifies image data for an [EmbeddedImage](#). This element MUST have a base64 ([\[RFC4648\]](#), section 4) string value. This element MUST be specified.

Following is the parent element of the **EmbeddedImage.ImageData** element.

Parent elements
EmbeddedImage

The following is the XML Schema definition of the **EmbeddedImage.ImageData** element.

```
<xsd:element name="ImageData" type="xsd:string">
```

### 2.306.3 EmbeddedImage.MIMETYPE

The **EmbeddedImage.MIMETYPE** element specifies the image format of an [EmbeddedImage](#). The **EmbeddedImage.MIMETYPE** element MUST be specified, its value MUST be a [ReportMIMETYPE](#), and the value MUST be specified only as a constant [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) value, not as an expression.

Following is the parent element of the **EmbeddedImage.MIMETYPE** element.

Parent element
EmbeddedImage

The following is the XML Schema definition of the **EmbeddedImage.MIMETYPE** element.

```
<xsd:element name="MIMETYPE" type="xsd:string">
```

## 2.307 ReportParameters

The **ReportParameters** element specifies an ordered list of [ReportParameter](#) elements for a [Report](#). This list MUST contain at least one **ReportParameter** element.

The following are the parent and child elements of the **ReportParameters** element.

Parent elements
Report

Child elements
<a href="#">ReportParameters.ReportParameter</a>

The following is the XML Schema definition of the **ReportParameters** element.

```
<xsd:complexType name="ReportParametersType" >
  <xsd:sequence>
    <xsd:element name="ReportParameter" type="ReportParameterType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.307.1 ReportParameters.ReportParameter

The **ReportParameters.ReportParameter** element specifies a [ReportParameter](#) in the collection of [ReportParameters](#) for a [Report](#). The **ReportParameters.ReportParameter** element MUST be specified. This element is of type **ReportParameter**.

Following is the parent element of the **ReportParameters.ReportParameter** element.

Parent elements
ReportParameters

The following is the XML Schema definition of the **ReportParameters.ReportParameter** element.

```
<xsd:element name="ReportParameter" type="ReportParameterType"
  maxOccurs="unbounded" />
```

## 2.308 ReportParameter

The **ReportParameter** element specifies information about a parameter to a [Report](#). Within a **ReportParameter**, the expression **User!Language** (rather than [Report.Language](#)) MUST be used for all language-dependent expressions and operations.

The following are the parent element, attribute, and child elements of the **ReportParameter** element.

Parent elements
Report

Attributes
<a href="#">ReportParameter.Name</a>

Child elements
<a href="#">ReportParameter.AllowBlank</a>
<a href="#">ReportParameter.DataType</a>
<a href="#">ReportParameter.DefaultValue</a>
<a href="#">ReportParameter.Hidden</a>
<a href="#">ReportParameter.MultiValue</a>
<a href="#">ReportParameter.Nullable</a>
<a href="#">ReportParameter.Prompt</a>
<a href="#">ReportParameter.UsedInQuery</a>
<a href="#">ReportParameter.ValidValues</a>

The following is the XML Schema definition of the **ReportParameter** element.

```

<xsd:complexType name="ReportParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataType">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Boolean" />
          <xsd:enumeration value="DateTime" />
          <xsd:enumeration value="Integer" />
          <xsd:enumeration value="Float" />
          <xsd:enumeration value="String" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Nullable" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DefaultValue" type="DefaultValueType" minOccurs="0" />
    <xsd:element name="AllowBlank" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Prompt" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ValidValues" type="ValidValuesType" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="MultiValue" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="UsedInQuery" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="False" />
          <xsd:enumeration value="True" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.308.1 ReportParameter.Name

The **ReportParameter.Name** attribute specifies a unique identifier for a [ReportParameter](#). This attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [UTR15] that is unique among all **report parameters** in the [Report](#).

Following is the parent element of the **ReportParameter.Name** attribute.

Parent elements
ReportParameter

The following is the XML Schema definition of the **ReportParameter.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.308.2 ReportParameter.AllowBlank

The **ReportParameter.AllowBlank** element specifies that an empty [String](#) ([XMLSCHEMA2/2] section 3.2.1) is allowed as a value for a [ReportParameter](#). The **ReportParameter.AllowBlank** element is ignored if the data type of the [ReportParameter.DataType](#) value is not **String**.

The **ReportParameter.AllowBlank** element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **ReportParameter.AllowBlank** element.

Parent elements
ReportParameter

The following is the XML Schema definition of the **ReportParameter.AllowBlank** element.

```
<xsd:element name="AllowBlank" type="xsd:boolean" minOccurs="0" />
```

### 2.308.3 ReportParameter.DataType

The **ReportParameter.DataType** element specifies the data type of a [ReportParameter](#). This element MUST be specified. The value of this element MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is one of the following:

**Boolean:** The value or values for [ReportParameter.ValidValues](#) and [ReportParameter.DefaultValue](#) and the value or values specified by the user for the **ReportParameter** are "Boolean".

**DateTime:** The value or values for **ReportParameter.ValidValues** and **ReportParameter.DefaultValue** and the value or values that are specified by the user for the **ReportParameter** are all "DateTime".

**Integer:** The value or values for **ReportParameter.ValidValues** and **ReportParameter.DefaultValue** and the value or values specified by the user for the **ReportParameter** are all "Integer".



**Float:** The value or values for **ReportParameter.ValidValues** and **ReportParameter.DefaultValue** and the value or values specified by the user for the **ReportParameter** are all "Float".

**String:** The value or values for **ReportParameter.ValidValues** and **ReportParameter.DefaultValue** and the value or values specified by the user for the **ReportParameter** are all "String".

Following is the parent element of the **ReportParameter.DataType** element.

Parent elements
ReportParameter

The following is the XML Schema definition of the **ReportParameter.DataType** element.

```
<xsd:element name="DataType">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Boolean" />
      <xsd:enumeration value="DateTime" />
      <xsd:enumeration value="Integer" />
      <xsd:enumeration value="Float" />
      <xsd:enumeration value="String" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

#### 2.308.4 ReportParameter.DefaultValue

The **ReportParameter.DefaultValue** element specifies the default value or values to use for a **report parameter** if values are not provided by the user. If a value is not provided as a part of the [ReportParameter](#) definition or by the user, the value of the **ReportParameter.DefaultValue** element is interpreted as NULL.

This element MUST be specified if the [ReportParameter.Prompt](#) element is not specified and either [ReportParameter.Nullable](#) is false or a [ReportParameter.ValidValues](#) list is provided that does not contain a **null value**.

The **ReportParameter.DefaultValue** element is of type [DefaultValue](#).

Following is the parent element of the **ReportParameter.DefaultValue** element.

Parent elements
ReportParameter

The following is the XML Schema definition of the **ReportParameter.DefaultValue** element.

```
<xsd:element name="DefaultValue" type="DefaultValueType" minOccurs="0" />
```

#### 2.308.5 ReportParameter.Hidden

**Applies to** [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **ReportParameter.Hidden** element specifies whether a [ReportParameter](#) is not displayed to the user at runtime. The **ReportParameter.Hidden** element is optional. If this element is present, its

value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **ReportParameter.Hidden** element.

Parent elements
ReportParameter

The following is the XML Schema definition of the **ReportParameter.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
```

### 2.308.6 ReportParameter.MultiValue

The **ReportParameter.MultiValue** element specifies whether a [ReportParameter](#) can take a set of values rather than a single value. The **ReportParameter.MultiValue** element is ignored for **report parameters** that have the [ReportParameter.DataType](#) set to a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2).

The **ReportParameter.MultiValue** element is optional. If this element is present, its value MUST be a **Boolean**. If this element is not present, its value is interpreted as false.

Following is the parent element of the **ReportParameter.MultiValue** element.

Parent elements
ReportParameter

The following is the XML Schema definition of the **ReportParameter.MultiValue** element.

```
<xsd:element name="MultiValue" type="xsd:boolean" minOccurs="0" />
```

### 2.308.7 ReportParameter.Nullable

The **ReportParameter.Nullable** element specifies whether the value of a [ReportParameter](#) can be null. If the value of the [ReportParameter.MultiValue](#) element is true, the value of the **ReportParameter.Nullable** element MUST NOT be true.

This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **ReportParameter.Nullable** element.

Parent elements
ReportParameter

The following is the XML Schema definition of the **ReportParameter.Nullable** element.

```
<xsd:element name="Nullable" type="xsd:boolean" minOccurs="0" />
```

## 2.308.8 ReportParameter.Prompt

The **ReportParameter.Prompt** element specifies the text to use when prompting the user to provide the value or values for a [ReportParameter](#). The **ReportParameter.Prompt** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1). If this element is not present, the user is not prompted for, or otherwise allowed to provide, a value for this parameter.

Following is the parent element of the **ReportParameter.Prompt** element.

Parent elements
ReportParameter

The following is the XML Schema definition of the **ReportParameter.Prompt** element.

```
<xsd:element name="Prompt" type="StringLocIDType" minOccurs="0" />
```

## 2.308.9 ReportParameter.UsedInQuery

The **ReportParameter.UsedInQuery** element specifies whether the [ReportParameter](#) is used in a [Query](#) in a [Report](#). This is necessary to determine whether queries MUST be re-executed if the **ReportParameter** value changes.

The **ReportParameter.UsedInQuery** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is one of the following:

**Auto** (default): The value of the **ReportParameter.UsedInQuery** element is interpreted as true if any [QueryParameter](#) value expression is a simple reference to this **ReportParameter**, if there is a [Subreport](#) in the report, or if there exists any **QueryParameter** value expression that is anything but a constant or a simple **ReportParameter** reference. Otherwise, the value of the **ReportParameter.UsedInQuery** element is interpreted as false.

**True**: The **ReportParameter** is used in a **query** in the report.

**False**: The **ReportParameter** is not used in any query in the report.

Following is the parent element of the **ReportParameter.UsedInQuery** element.

Parent elements
ReportParameter

The following is the XML Schema definition of the **ReportParameter.UsedInQuery** element.

```
<xsd:element name="UsedInQuery" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="False" />
      <xsd:enumeration value="True" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.308.10 ReportParameter.ValidValues

The **ReportParameter.ValidValues** element specifies the possible values that can be used for the [ReportParameter](#) element. The **ReportParameter.ValidValues** element is optional. This element is of type [ValidValues](#).

Following is the parent element of the **ReportParameter.ValidValues** element.

Parent elements
ReportParameter

The following is the XML Schema definition of the **ReportParameter.ValidValues** element.

```
<xsd:element name="ValidValues" type="ValidValuesType" minOccurs="0" />
```

## 2.309 ReportParametersLayout

**Applies to** [RDL 2016/01](#)

The **ReportParametersLayout** element specifies information for a report parameter layout for a [Report](#).

The following are the parent and child elements of the **ReportParametersLayout** element.

Parent elements
Report

Child elements
<a href="#">ReportParametersLayout.GridLayoutDefinition</a>

The following is the XML Schema definition of the **ReportParametersLayout** element.

```
<xsd:complexType name="ReportParametersLayoutType">  
  <xsd:choice>  
    <xsd:element name="GridLayoutDefinition" type="GridLayoutDefinitionType" minOccurs="1" />  
  </xsd:choice>  
</xsd:complexType>
```

### 2.309.1 ReportParametersLayout.GridLayoutDefinition

**Applies to** [RDL 2016/01](#)

The **ReportParametersLayout.GridLayoutDefinition** element specifies information about a **report parameter layout grid** for a [Report](#). The **ReportParametersLayout.GridLayoutDefinition** element **MUST** be specified. This element is of type **GridLayoutDefinition**.

Following is the parent element of the **ReportParametersLayout.GridLayoutDefinition** element.

<b>Parent elements</b>
<a href="#">ReportParametersLayout</a>

The following is the XML Schema definition of the **ReportParametersLayout.GridLayoutDefinition** element.

```
<xsd:element name="GridLayoutDefinition" type="GridLayoutDefinitionType" minOccurs="1" />
```

## 2.310 GridLayoutDefinition

**Applies to** [RDL 2016/01](#)

The **GridLayoutDefinition** element specifies information about a **report parameter layout grid** for a [Report](#).

The following are the parent and child elements of the **GridLayoutDefinition** element.

<b>Parent elements</b>
<a href="#">ReportParametersLayout</a>

<b>Child elements</b>
<a href="#">GridLayoutDefinition.NumberOfColumns</a>
<a href="#">GridLayoutDefinition.NumberOfRows</a>
<a href="#">GridLayoutDefinition.CellDefinitions</a>

The following is the XML Schema definition of the **GridLayoutDefinition** element.

```
<xsd:complexType name="GridLayoutDefinitionType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="NumberOfColumns" type="xsd:int" minOccurs="1" />
    <xsd:element name="NumberOfRows" type="xsd:int" minOccurs="1" />
    <xsd:element name="CellDefinitions" type="CellDefinitionsType" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

### 2.310.1 GridLayoutDefinition.NumberOfColumns

**Applies to** [RDL 2016/01](#)

The **GridLayoutDefinition.NumberOfColumns** element specifies the number of columns for a **report parameter layout grid**. This element **MUST** be specified. Its value **MUST** be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) that is greater than or equal to 1 and less than or equal to 2147483647. [<61>](#)

Following is the parent element of the **GridLayoutDefinition.NumberOfColumns** element.

<b>Parent elements</b>
<a href="#">GridLayoutDefinition</a>

The following is the XML Schema definition of the **GridLayoutDefinition.NumberOfColumns** element.

```
<xsd:element name="NumberOfColumns" type="xsd:int" minOccurs="1" />
```

### 2.310.2 GridLayoutDefinition.NumberOfRows

*Applies to [RDL 2016/01](#)*

The **GridLayoutDefinition.NumberOfRows** element specifies the number of rows for a **report parameter layout grid**. This element **MUST** be specified. Its value **MUST** be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) that is greater than or equal to 1 and less than or equal to 2147483647. [<62>](#)

Following is the parent element of the **GridLayoutDefinition.NumberOfRows** element.

Parent elements
<a href="#">GridLayoutDefinition</a>

The following is the XML Schema definition of the **GridLayoutDefinition.NumberOfRows** element.

```
<xsd:element name="NumberOfRows" type="xsd:int" minOccurs="1" />
```

### 2.310.3 GridLayoutDefinition.CellDefinitions

*Applies to [RDL 2016/01](#)*

The **GridLayoutDefinition.CellDefinitions** element specifies cell definitions for a **report parameter layout grid**. The **GridLayoutDefinition.CellDefinitions** element is optional. This element is of type **CellDefinitions**.

Following is the parent element of the **GridLayoutDefinition.CellDefinitions** element.

Parent elements
<a href="#">GridLayoutDefinition</a>

The following is the XML Schema definition of the **GridLayoutDefinition.CellDefinitions** element.

```
<xsd:element name="CellDefinitions" type="CellDefinitionsType" minOccurs="0" />
```

### 2.311 CellDefinitions

*Applies to [RDL 2016/01](#)*

The **CellDefinitions** element specifies an ordered list of **CellDefinition** elements for a **report parameter layout grid**. This list **MUST** contain at least one **CellDefinition** element.

The following are the parent and child elements of the **CellDefinitions** element.

Parent elements
<a href="#">GridLayoutDefinition</a>

<b>Child elements</b>
-----------------------

<a href="#">CellDefinitions.CellDefinition</a>
--

The following is the XML Schema definition of the **CellDefinition** element.

```
<xsd:complexType name="CellDefinitionsType" >
  <xsd:sequence>
    <xsd:element name="CellDefinition" type="CellDefinitionType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

### 2.311.1 CellDefinitions.CellDefinition

**Applies to** [RDL 2016/01](#)

The **CellDefinitions.CellDefinition** element specifies a cell definition for a **report parameter layout grid**. The **CellDefinitions.CellDefinition** element MUST be specified. This element is of type **CellDefinition**.

The quantity of **CellDefinitions.CellDefinition** elements MUST match the quantity of [ReportParameters.ReportParameter](#) elements.

There MUST NOT be any **CellDefinitions.CellDefinition** elements that specify the same combination of [CellDefinition.ColumnIndex](#) and [CellDefinition.RowIndex](#) or reference the same [CellDefinition.ParameterName](#).

An empty row in the report parameter layout grid is a row that does not have any **CellDefinitions.CellDefinition** elements referencing it with the **CellDefinition.RowIndex** element. [<63>](#)

Following is the parent element of the **CellDefinitions.CellDefinition** element.

<b>Parent elements</b>
------------------------

<a href="#">CellDefinitions</a>
---------------------------------

The following is the XML Schema definition of the **CellDefinitions.CellDefinition** element.

```
<xsd:element name="CellDefinition" type="CellDefinitionType" minOccurs="1"
maxOccurs="unbounded" />
```

### 2.312 CellDefinition

**Applies to** [RDL 2016/01](#)

The **CellDefinition** element specifies information about a cell definition for a **report parameter layout grid**.

The following are the parent and child elements of the **CellDefinition** element.

<b>Parent elements</b>
<a href="#">CellDefinitions</a>

<b>Child elements</b>
<a href="#">CellDefinition.ColumnIndex</a>
<a href="#">CellDefinition.RowIndex</a>
<a href="#">CellDefinition.ParameterName</a>

The following is the XML Schema definition of the **CellDefinition** element.

```
<xsd:complexType name="CellDefinitionType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="ColumnIndex" type="xsd:int" minOccurs="1" />
    <xsd:element name="RowIndex" type="xsd:int" minOccurs="1" />
    <xsd:element name="ParameterName" type="xsd:normalizedString" minOccurs="1" />
  </xsd:choice>
</xsd:complexType>
```

### 2.312.1 CellDefinition.ColumnIndex

*Applies to* [RDL 2016/01](#)

The **CellDefinition.ColumnIndex** element specifies the column index for a cell within a **report parameter layout grid**. This element MUST be specified. Its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) that is greater than or equal to 0 and less than the value of the [GridLayoutDefinition.NumberOfColumns](#) element.

Following is the parent element of the **CellDefinition.ColumnIndex** element.

<b>Parent elements</b>
<a href="#">CellDefinition</a>

The following is the XML Schema definition of the **CellDefinition.ColumnIndex** element.

```
<xsd:element name="ColumnIndex" type="xsd:int" minOccurs="1" />
```

### 2.312.2 CellDefinition.RowIndex

*Applies to* [RDL 2016/01](#)

The **CellDefinition.RowIndex** element specifies the row index for a cell within a **report parameter layout grid**. This element MUST be specified. Its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) that is greater than or equal to 0 and less than the value of the [GridLayoutDefinition.NumberOfRows](#) element.

Following is the parent element of the **CellDefinition.RowIndex** element.



<b>Parent elements</b>
<a href="#">CellDefinition</a>

The following is the XML Schema definition of the **CellDefinition.RowIndex** element.

```
<xsd:element name="RowIndex" type="xsd:int" minOccurs="1" />
```

### 2.312.3 CellDefinition.ParameterName

*Applies to [RDL 2016/01](#)*

The **CellDefinition.ParameterName** element specifies the referenced parameter for a cell definition. This element **MUST** be specified. Its value **MUST** equal one of the [ReportParameter.Name](#) attribute values.

Following is the parent element of the **CellDefinition.ParameterName** element.

<b>Parent elements</b>
<a href="#">CellDefinition</a>

The following is the XML Schema definition of the **CellDefinition.ParameterName** element.

```
<xsd:element name="ParameterName" type="xsd:normalizedString" minOccurs="1" />
```

### 2.313 DefaultValue

The **DefaultValue** element specifies the default values for the [ReportParameter](#) element. The **DefaultValue** element **MUST** specify either the [DefaultValue.Values](#) or the [DefaultValue.DataSetReference](#) element. If one of the values that is either specified in the **DefaultValue.Values** element or retrieved through the **DefaultValue.DataSetReference** is not valid, the entire set of default values **MUST** be treated as not valid.

The following are the parent and child elements of the **DefaultValue** element.

<b>Parent elements</b>
<a href="#">ReportParameter</a>

<b>Child elements</b>
<a href="#">DefaultValue.DataSetReference</a>
<a href="#">DefaultValue.Values</a>

The following is the XML Schema definition of the **DefaultValue** element.

```
<xsd:complexType name="DefaultValueType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataSetReference" type="DataSetReferenceType"
      minOccurs="0" />
    <xsd:element name="Values" type="ValuesType" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```

```

    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.313.1 DefaultValue.DataSetReference

The **DefaultValue.DataSetReference** element specifies the [DataSet](#) to use to obtain the default value or values for the [ReportParameter](#) element. If [DefaultValue.Values](#) is not specified, the **DefaultValue.DataSetReference** element MUST be specified.

If the **DefaultValue.DataSetReference** element is present and the value of the [ReportParameter.MultiValue](#) element is false, the default value MUST be the first value of the field that is specified in the [DataSetReference.ValueField](#) element. If the value of the **ReportParameter.MultiValue** element is true, all values of the field that is specified in the **DataSetReference.ValueField** element MUST be included.

The **DefaultValue.DataSetReference** element is of type [DataSetReference](#).

Following is the parent element of the **DefaultValue.DataSetReference** element.

Parent elements
<a href="#">DefaultValue</a>

The following is the XML Schema definition of the **DefaultValue.DataSetReference** element.

```

<xsd:element name="DataSetReference" type="DataSetReferenceType" minOccurs="0" />

```

### 2.313.2 DefaultValue.Values

The **DefaultValue.Values** element specifies the default values to use for the [ReportParameter](#) element. If [DefaultValue.DataSetReference](#) is not specified, the **DefaultValue.Values** element MUST be specified. This element is of type [Values](#).

Following is the parent element of the **DefaultValue.Values** element.

Parent elements
<a href="#">DefaultValue</a>

The following is the XML Schema definition of the **DefaultValue.Values** element.

```

<xsd:element name="Values" type="ValuesType" minOccurs="0"/>

```

## 2.314 DataSetReference

The **DataSetReference** element specifies the [DataSet](#) to use to obtain a list of values and, optionally, labels for the [ValidValues](#) or the [DefaultValue](#) elements of a [ReportParameter](#).

The following are the parent and child elements of the **DataSetReference** element.

Parent elements
DefaultValue
ValidValues

Child elements
<a href="#">DataSetReference.DataSetName</a>
<a href="#">DataSetReference.LabelField</a>
<a href="#">DataSetReference.ValueField</a>

The following is the XML Schema definition of the **DataSetReference** element.

```
<xsd:complexType name="DataSetReferenceType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" />
    <xsd:element name="ValueField" type="xsd:string" />
    <xsd:element name="LabelField" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.314.1 DataSetReference.DataSetName

The **DataSetReference.DataSetName** element specifies the name of the [DataSet](#) that is being referenced. This element **MUST** be specified. The value of this element **MUST** be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1).

Following is the parent element for the **DataSetReference.DataSetName** element.

Parent elements
<a href="#">DataSetReference</a>

The following is the XML Schema definition of the **DataSetReference.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" />
```

### 2.314.2 DataSetReference.LabelField

The **DataSetReference.LabelField** element specifies the name of the field in the referenced [DataSet](#) from which values are retrieved to populate the labels of a parameter's [ValidValues](#). Labels are used as the values to display to the user for the parameter value selection. The **DataSetReference.LabelField** element is optional.

If this element is present, its value **MUST** be a [String](#) ([XMLSCHEMA2/2](#) section 3.2.1). If this element is not present or if it returns a **null value**, the values that are populated with the **ValueField** element **MUST** be used.

The **DataSetReference.LabelField** element is ignored when the parent of the [DataSetReference](#) is the [DefaultValue](#) element.

Following is the parent element of the **DataSetReference.LabelField** element.

Parent elements
<a href="#">DataSetReference</a>

The following is the XML Schema definition of the **DataSetReference.LabelField** element.

```
<xsd:element name="LabelField" type="xsd:string" minOccurs="0" />
```

### 2.314.3 DataSetReference.ValueField

The **DataSetReference.ValueField** element specifies the name of the field in the referenced [DataSet](#) from which values are retrieved to populate the values of a parameter's [ValidValues](#) or [DefaultValue](#). The **DataSetReference.ValueField** element MUST be specified. The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Following is the parent element of the **DataSetReference.ValueField** element.

Parent elements
<a href="#">DataSetReference</a>

The following is the XML Schema definition of the **DataSetReference.ValueField** element.

```
<xsd:element name="ValueField" type="xsd:string" />
```

### 2.315 Values

The **Values** element specifies a set of values. For a non-multivalued parameter (such as when [ReportParameter.MultiValue](#) is set to "false"), this collection MUST contain one **Value** element.

The following are the parent and child elements of the **Values** element.

Parent elements
<a href="#">DefaultValue</a>

Child elements
<a href="#">Values.Value</a>

The following is the XML Schema definition of the **Values** element.

```
<xsd:complexType name="ValuesType">
  <xsd:sequence>
    <xsd:element name="Value" type="xsd:string" minOccurs="1"
      maxOccurs="unbounded" nullable="true" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

</xsd:complexType>

### 2.315.1 Values.Value

The **Values.Value** element specifies the value or expression to evaluate to obtain the value for the [DefaultValue](#) for a [ReportParameter](#). The **Values.Value** element MUST be specified.

If the value of this element is an expression, the expression MUST NOT refer to fields, to report items, or to any report parameters that occur after this **ReportParameter** in the [ReportParameters](#) collection.

If the expression returns an array, each item in the array is treated as a single value. Items in the array MUST NOT be arrays. If the **ReportParameter** element has [ReportParameter.MultiValue](#) set to false, the first item in the array MUST be used.

At least one **Value** element MUST be specified for a [Values](#) collection. If the value of the **ReportParameter.MultiValue** element is set to false, the **Value** element MUST be specified exactly once for a **Values** collection. NULL is a valid value for the **Values.Value** element.

Following is the parent element of the **Values.Value** element.

Parent elements
Values

The following is the XML Schema definition of the **Values.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1"
maxOccurs="unbounded" nullable="true"/>
```

### 2.316 ValidValues

The **ValidValues** element specifies the possible values for a [ReportParameter](#) and for populating UI selection lists for users to select a parameter value. This element is optional. If this element is present, it MUST specify either [ValidValues.DataSetReference](#) or [ValidValues.ParameterValues](#).

The following are the parent and child elements of the **ValidValues** element.

Parent elements
ReportParameter

Child elements
ValidValues.DataSetReference
ValidValues.ParameterValues

The following is the XML Schema definition of the **ValidValues** element.

```
<xsd:complexType name="ValidValuesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
```

```

<xsd:element name="DataSetReference" type="DataSetReferenceType"
  minOccurs="0" />
<xsd:element name="ParameterValues" type="ParameterValuesType"
  minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.316.1 ValidValues.DataSetReference

The **ValidValues.DataSetReference** element specifies the [DataSet](#) in a [Report](#). The **ValidValues.DataSetReference** element also specifies the fields from that dataset to use to obtain a list of values and, optionally, labels for use in value validation and for populating the UI for the [ReportParameter](#).

The **ValidValues.DataSetReference** element MUST be specified if [ValidValues.ParameterValues](#) is not specified. The **ValidValues.DataSetReference** element is of type [DataSetReference](#).

Following is the parent element of the **ValidValues.DataSetReference** element.

Parent elements
<a href="#">ValidValues</a>

The following is the XML Schema definition of the **ValidValues.DataSetReference** element.

```

<xsd:element name="DataSetReference" type="DataSetReferenceType" minOccurs="0" />

```

### 2.316.2 ValidValues.ParameterValues

The **ValidValues.ParameterValues** element specifies the list of values and, optionally, labels for use in value validation and for populating the UI for the [ReportParameter](#). If [ValidValues.DataSetReference](#) is not specified, the **ValidValues.ParameterValues** element MUST be specified. This element is of type [ParameterValues](#).

Following is the parent element of the **ValidValues.ParameterValues** element.

Parent elements
<a href="#">ValidValues</a>

The following is the XML Schema definition of the **ValidValues.ParameterValues** element.

```

<xsd:element name="ParameterValues" type="ParameterValuesType" minOccurs="0" />

```

## 2.317 ParameterValues

The **ParameterValues** element specifies an ordered list of [ParameterValue](#) elements that are used to populate UI selection lists for users to select a value for a [Parameter](#).

The following are the parent and child elements of the **ParameterValues** element.

<b>Parent elements</b>
<a href="#">ValidValues</a>

<b>Child elements</b>
<a href="#">ParameterValues.ParameterValue</a>

The following is the XML Schema definition of the **ParameterValues** element.

```
<xsd:complexType name="ParameterValuesType" >
  <xsd:sequence>
    <xsd:element name="ParameterValue" type="ParameterValueType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.317.1 ParameterValues.ParameterValue

The **ParameterValues.ParameterValue** element specifies a possible value for a [Parameter](#). At least one **ParameterValues.ParameterValue** element MUST be specified for a [ParameterValues](#) collection. The **ParameterValues.ParameterValue** element is of type [ParameterValue](#).

Following is the parent element of the **ParameterValues.ParameterValue** element.

<b>Parent elements</b>
<a href="#">ParameterValues</a>

The following is the XML Schema definition of the **ParameterValues.ParameterValue** element.

```
<xsd:element name="ParameterValue" type="ParameterValueType" maxOccurs="unbounded">
```

### 2.318 ParameterValue

The **ParameterValue** element specifies a value/label pair for [ValidValues](#).

The following are the parent and child elements of the **ParameterValue** element.

<b>Parent elements</b>
<a href="#">ParameterValues</a>

<b>Child elements</b>
<a href="#">ParameterValue.Label</a>
<a href="#">ParameterValue.Value</a>

The following is the XML Schema definition of the **ParameterValue** element.

```
<xsd:complexType name="ParameterValueType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="StringLocIDType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

### 2.318.1 ParameterValue.Label

The **ParameterValue.Label** element specifies the text to use to describe the [ParameterValue.Value](#) to display in a parameter drop-down at runtime. The **ParameterValue.Label** element is optional.

If the **ParameterValue.Label** element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**. If this element is not present, the **ParameterValue.Value** is applied as the label text. If **ParameterValue.Value** is also not present, an empty string is used.

If the **ParameterValue.Value** expression returns an array, the expression for the label MUST also return an array with the same number of items. If the **ParameterValue.Value** expression does not return an array, the expression for the label expression MUST NOT return an array.

Following is the parent element of the **ParameterValue.Label** element.

Parent elements
<a href="#">ParameterValue</a>

The following is the XML Schema definition of the **ParameterValue.Label** element.

```
<xsd:element name="Label" type="StringLocIDType" minOccurs="0" />
```

### 2.318.2 ParameterValue.Value

The **ParameterValue.Value** element specifies the value for a [Parameter](#). This element is optional. The value of the **ParameterValue.Value** element MUST be a **Variant** or an expression that evaluates to a **Variant**. If this element is not present, its value is interpreted as null.

If the **ParameterValue.Value** expression returns an array, each item in the array MUST be treated as a single value. The items in the array MUST NOT be arrays.

Following is the parent element of the **ParameterValue.Value** element.

Parent elements
<a href="#">ParameterValue</a>

The following is the XML Schema definition of the **ParameterValue.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="0"/>
```



## 2.319 Variables

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Variables** element specifies a series of named expressions to be evaluated within a group or a report. This element is optional.

The following are the parent and child elements of the **Variables** element.

Parent elements
<a href="#">Group</a>
<a href="#">Report</a>

Child elements
<a href="#">Variables.Variable</a>

The following is the XML Schema definition of the **Variables** element in RDL 2008/01.

```
<xsd:complexType name="VariablesType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Variable" type="VariableType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **Variables** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="VariablesType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Variable" type="VariableType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.319.1 Variables.Variable

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

The **Variables.Variable** element specifies a named expression to be evaluated, used for a group or [Report](#), and made available in the [Variables global collection](#). The **Variables.Variable** element is of type [Variable](#). This element MUST be specified at least once within a **Variables** collection.

Following is the parent element of the **Variables.Variable** element.

Parent elements
<a href="#">Variables</a>

The following is the XML Schema definition of the **Variables.Variable** element.

```
<xsd:element name="Variable" type="VariableType" minOccurs="1" maxOccurs="unbounded" />
```

## 2.320 Variable

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Variable** element specifies a named expression to be evaluated and used within a group or report. This element **MUST** be specified at least once within a [Variables](#) collection.

The following are the parent elements, attributes, and child elements of the **Variable** element.

Parent elements
<a href="#">Variables</a>

Attributes
<a href="#">Variable.Name</a>

The following are the child elements of the **Variable** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Child elements
<a href="#">Variable.Value</a>
<a href="#">Variable.Value.Datatype</a>

The following are the child elements of the **Variable** element in RDL 2010/01 and RDL 2016/01.

Child elements
<a href="#">Variable.Writable</a>

The following is the XML Schema definition of the **Variable** element in RDL 2008/01.

```
<xsd:complexType name="VariableType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="StringWithDataTypeAttribute" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **Variable** element in RDL 2010/01 and RDL 2016/01.

```
<xsd:complexType name="VariableType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="StringWithDataTypeAttribute" minOccurs="1"
      maxOccurs="1" />
  </xsd:choice>
</xsd:complexType>
```

```

    <xsd:element name="Writable" type="xsd:boolean" minOccurs="0" maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.320.1 Variable.Name

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Variable.Name** attribute specifies a unique name for a [Variable](#) to be used in expressions in a report. This attribute **MUST** be specified. The value of this attribute **MUST** be a case-sensitive **CLS-compliant identifier** [UTR15] that is unique for all the **Variable** elements within a [Report](#).

Following is the parent element of the **Variable.Name** attribute.

Parent elements
<a href="#">Variable</a>

The following is the XML Schema definition of the **Variable.Name** attribute.

```

<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />

```

### 2.320.2 Variable.Value

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Variable.Value** element specifies an expression to evaluate globally for a [Report](#) or for instances of a particular group. This element **MUST** be specified.

The following are the parent element and attributes of the **Variable.Value** element.

Parent elements
<a href="#">Variable</a>

Attributes
<a href="#">Variable.Value.DataType</a>

The following is the XML Schema definition of the **Variable.Value** element.

```

<xsd:element name="Value" type="StringWithDataTypeAttribute"
  minOccurs="1" maxOccurs="1">

```

### 2.320.3 Variable.Value.DataType

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Variable.Value.DataType** attribute specifies the data type of the [Variable.Value](#) element if the value is a constant. This attribute is optional. If this attribute is present, its value **MUST** be a [String](#)

([\[XMLSCHEMA2/2\]](#) section 3.2.1), a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2), a [DateTime](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.7), an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17), or a [Float](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.4). If this attribute is not present, its value is interpreted as a **String**.

Following is the parent element of the **Variable.Value.DataType** attribute.

Parent elements
Variable.Value

#### 2.320.4 Variable.Writable

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

The **Variable.Writable** element specifies whether the [Report](#) variable is writable by using an expression or custom code. This element is optional and **MUST NOT** be specified if the parent of the [Variable](#) element is a [Group](#).

If this element is present, its value **MUST** be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **Variable.Writable** element.

Parent elements
Variable

The following is the XML Schema definition of the **Variable.Writable** element.

```
<xsd:element name="Writable" type="xsd:boolean" minOccurs="0" maxOccurs="1" />
```

#### 2.321 DataSources

The **DataSources** element specifies an ordered list of [DataSource](#) elements for a [Report](#).

The following are the parent and child elements of the **DataSources** element.

Parent elements
Report

Child elements
<a href="#">DataSources.DataSource</a>

The following is the XML Schema definition of the **DataSources** element.

```
<xsd:complexType name="DataSourcesType">
  <xsd:sequence>
    <xsd:element name="DataSource" type="DataSourceType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

```
</xsd:complexType>
```

### 2.321.1 DataSources.DataSource

The **DataSources.DataSource** element specifies a data source in the collection of [DataSources](#) for a [Report](#). This element is of type [DataSource](#).

Following is the parent element of the **DataSources.DataSource** element.

Parent elements
<a href="#">DataSources</a>

The following is the XML Schema definition of the **DataSources.DataSource** element.

```
<xsd:element name="DataSource" type="DataSourceType" maxOccurs="unbounded" />
```

### 2.322 DataSource

The **DataSource** element specifies information about a data source. This element MUST include either the [DataSource.DataSourceReference](#) element or the [DataSource.ConnectionProperties](#) element.

The following are the parent element, attribute, and child elements of the **DataSource** element.

Parent elements
<a href="#">DataSources</a>

Attributes
<a href="#">DataSource.Name</a>

Child elements
<a href="#">DataSource.ConnectionProperties</a>
<a href="#">DataSource.DataSourceReference</a>
<a href="#">DataSource.Transaction</a>

The following is the XML Schema definition of the **DataSource** element.

```
<xsd:complexType name="DataSourceType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Transaction" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ConnectionProperties" type="ConnectionPropertiesType"
      minOccurs="0" />
    <xsd:element name="DataSourceReference" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

</xsd:complexType>

### 2.322.1 DataSource.Name

The **DataSource.Name** attribute specifies a unique identifier for a data source. This attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [UTR15]. The **DataSource.Name** attribute MUST be unique among all data sources within a [Report](#).

Following is the parent element of the **DataSource.Name** attribute.

Parent elements
<a href="#">DataSource</a>

The following is the XML Schema definition of the **DataSource.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:string" use="required" />
```

### 2.322.2 DataSource.ConnectionProperties

The **DataSource.ConnectionProperties** element specifies information about how to connect to a data source. This element is optional. If this element is present, the [DataSource.DataSourceReference](#) element MUST be omitted. The **DataSource.ConnectionProperties** element is of type [ConnectionProperties](#).

Following is the parent element of the **DataSource.ConnectionProperties** element.

Parent elements
<a href="#">DataSource</a>

The following is the XML Schema definition of the **DataSource.ConnectionProperties** element.

```
<xsd:element name="ConnectionProperties" type="ConnectionPropertiesType"
  minOccurs="0" />
```

### 2.322.3 DataSource.DataSourceReference

The **DataSource.DataSourceReference** element specifies the absolute path (such as "/salesreports/salesdatabase") or the relative path (such as "salesdatabase") to a shared data source on the current report server. Relative paths start in the same folder as the referencing report.

This element is optional. If this element is present, the [DataSource.ConnectionProperties](#) element MUST be omitted.

Following is the parent element of the **DataSource.DataSourceReference** element.

Parent elements
<a href="#">DataSource</a>

The following is the XML Schema definition of the **DataSource.DataSourceReference** element.

```
<xsd:element name="DataSourceReference" type="xsd:string" minOccurs="0" />
```

#### 2.322.4 DataSource.Transaction

The **DataSource.Transaction** element specifies whether the datasets that use this data source are executed in a single transaction. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false.

Following is the parent element of the **DataSource.Transaction** element.

Parent elements
<a href="#">DataSource</a>

The following is the XML Schema definition of the **DataSource.Transaction** element.

```
<xsd:element name="Transaction" type="xsd:boolean" minOccurs="0" />
```

#### 2.323 ConnectionProperties

The **ConnectionProperties** element specifies information about how to connect to a data source.

The following are the parent and child elements of the **ConnectionProperties** element.

Parent elements
<a href="#">DataSource</a>

Child elements
<a href="#">ConnectionProperties.ConnectionString</a>
<a href="#">ConnectionProperties.DataProvider</a>
<a href="#">ConnectionProperties.IntegratedSecurity</a>
<a href="#">ConnectionProperties.Prompt</a>

The following is the XML Schema definition of the **ConnectionProperties** element.

```
<xsd:complexType name="ConnectionPropertiesType">  
  <xsd:choice minOccurs="1" maxOccurs="unbounded">  
    <xsd:element name="DataProvider" type="xsd:string" />  
    <xsd:element name="ConnectionString" type="xsd:string" />  
    <xsd:element name="IntegratedSecurity" type="xsd:boolean" minOccurs="0" />  
    <xsd:element name="Prompt" type="StringLocIDType" minOccurs="0" />  
    <xsd:any namespace="##other" processContents="skip" />  
  </xsd:choice>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

### 2.323.1 ConnectionProperties.ConnectionString

The **ConnectionProperties.ConnectionString** element specifies the information necessary to connect to a **data provider**. This element MUST be specified. The value of this element MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **ConnectionProperties.ConnectionString** element.

Parent elements
<a href="#">ConnectionProperties</a>

The following is the XML Schema definition of the **ConnectionProperties.ConnectionString** element.

```
<xsd:element name="ConnectionString" type="xsd:string" />
```

### 2.323.2 ConnectionProperties.DataProvider

The **ConnectionProperties.DataProvider** element specifies the name for the type of a **data provider** to use for a data source. This element MUST be specified. The value of this element MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1).

Following is the parent element of the **ConnectionProperties.DataProvider** element.

Parent elements
<a href="#">ConnectionProperties</a>

The following is the XML Schema definition of the **ConnectionProperties.DataProvider** element.

```
<xsd:element name="DataProvider" type="xsd:string" />
```

### 2.323.3 ConnectionProperties.IntegratedSecurity

The **ConnectionProperties.IntegratedSecurity** element specifies that the connection to a data source connects by using integrated security. This element is optional. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2).

Following is the parent element of the **ConnectionProperties.IntegratedSecurity** element.

Parent elements
<a href="#">ConnectionProperties</a>

The following is the XML Schema definition of the **ConnectionProperties.IntegratedSecurity** element.

```
<xsd:element name="IntegratedSecurity" type="xsd:boolean" minOccurs="0" />
```

### 2.323.4 ConnectionProperties.Prompt

The **ConnectionProperties.Prompt** element specifies the text to display to the user when prompting for credentials for a data source. This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1).



Following is the parent element of the **ConnectionProperties.Prompt** element.

Parent elements
<a href="#">ConnectionProperties</a>

The following is the XML Schema definition of the **ConnectionProperties.Prompt** element.

```
<xsd:element name="Prompt" type="StringLocIDType" minOccurs="0" />
```

## 2.324 DataSets

The **DataSets** element contains an ordered list of [DataSet](#) elements for a [Report](#). If the value of the [DataSource.Transaction](#) property of the [DataSource](#) that is referenced in the [DataSet.Query](#) is true, the execution order **MUST** follow the order of the **datasets** as they are specified in the report for all datasets that use the same **DataSource**.

The following are the parent and child elements of the **DataSets** element.

Parent elements
Report

Child elements
<a href="#">DataSets.DataSet</a>

The following is the XML Schema definition of the **DataSets** element.

```
<xsd:complexType name="DataSetsType">  
  <xsd:sequence>  
    <xsd:element name="DataSet" type="DataSetType" maxOccurs="unbounded" />  
  </xsd:sequence>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

### 2.324.1 DataSets.DataSet

The **DataSets.DataSet** element specifies a [DataSet](#) in the collection of [DataSets](#) for a [Report](#). The **DataSets.DataSet** element is of type **DataSet**.

Following is the parent element of the **DataSets.DataSet** element.

Parent elements
DataSets

The following is the XML Schema definition of the **DataSets.DataSet** element.

```
<xsd:element name="DataSet" type="DataSetType" maxOccurs="unbounded" />
```

## 2.325 DataSet

The **DataSet** element specifies information about a set of data to be used as a part of a [Report](#).

The following are the parent elements, attributes, and child elements of the **DataSet** element.

Parent elements
<a href="#">DataSets</a>

Attributes
<a href="#">DataSet.Name</a>

Child elements
<a href="#">DataSet.AccentSensitivity</a>
<a href="#">DataSet.CaseSensitivity</a>
<a href="#">DataSet.Collation</a>
<a href="#">DataSet.Fields</a>
<a href="#">DataSet.Filters</a>
<a href="#">DataSet.InterpretSubtotalsAsDetails</a>
<a href="#">DataSet.KanatypeSensitivity</a>
<a href="#">DataSet.Query</a>
<a href="#">DataSet.SharedDataSet</a>
<a href="#">DataSet.WidthSensitivity</a>

**Applies to** [RDL 2011/01](#)

Child elements
<a href="#">DataSet.DefaultRelationships</a>
<a href="#">DataSet.NullsAsBlanks</a>
<a href="#">DataSet.CollationCulture</a>

The following is the XML Schema definition of the **DataSet** element in [RDL 2003/10](#) and [RDL 2005/01](#).

```
<xsd:complexType name="DataSetType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Fields" type="FieldsType" minOccurs="0" />
    <xsd:element name="Query" type="QueryType" />
    <xsd:element name="CaseSensitivity" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="True" />
          <xsd:enumeration value="False" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>
```

```

        <xsd:enumeration value="Auto" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="Collation" type="xsd:string" minOccurs="0" />
<xsd:element name="AccentSensitivity" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="True" />
            <xsd:enumeration value="False" />
            <xsd:enumeration value="Auto" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="KanatypeSensitivity" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="True" />
            <xsd:enumeration value="False" />
            <xsd:enumeration value="Auto" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="WidthSensitivity" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="True" />
            <xsd:enumeration value="False" />
            <xsd:enumeration value="Auto" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **DataSet** element in [RDL 2008/01](#).

```

<xsd:complexType name="DataSetType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Fields" type="FieldsType" minOccurs="0" />
        <xsd:element name="Query" type="QueryType" />
        <xsd:element name="CaseSensitivity" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="True" />
                    <xsd:enumeration value="False" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Collation" type="xsd:string" minOccurs="0" />
        <xsd:element name="AccentSensitivity" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="True" />
                    <xsd:enumeration value="False" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="KanatypeSensitivity" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">

```

```

        <xsd:enumeration value="True" />
        <xsd:enumeration value="False" />
        <xsd:enumeration value="Auto" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="WidthSensitivity" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="True" />
            <xsd:enumeration value="False" />
            <xsd:enumeration value="Auto" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:element name="InterpretSubtotalsAsDetails" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="True" />
            <xsd:enumeration value="False" />
            <xsd:enumeration value="Auto" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **DataSet** element in [RDL 2010/01](#) and [RDL 2016/01](#).

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```

<xsd:complexType name="DataSetType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Fields" type="FieldsType" minOccurs="0" />
        <xsd:element name="Query" type="QueryType" minOccurs="0" />
        <xsd:element name="SharedDataSet" type="SharedDataSetType" minOccurs="0" />
        <xsd:element name="CaseSensitivity" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="True" />
                    <xsd:enumeration value="False" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Collation" type="xsd:string" minOccurs="0" />
        <xsd:element name="AccentSensitivity" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="True" />
                    <xsd:enumeration value="False" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="KanatypeSensitivity" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="True" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:choice>
</xsd:complexType>

```

```

        <xsd:enumeration value="False" />
        <xsd:enumeration value="Auto" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="WidthSensitivity" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="True" />
            <xsd:enumeration value="False" />
            <xsd:enumeration value="Auto" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:element name="InterpretSubtotalsAsDetails" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="True" />
            <xsd:enumeration value="False" />
            <xsd:enumeration value="Auto" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.325.1 DataSet.Name

The **DataSet.Name** attribute specifies a unique identifier for a [DataSet](#). This attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [UTR15].

The value of the **DataSet.Name** attribute MUST be unique among all datasets, data regions, and groups in the report.

Following is the parent element of the **DataSet.Name** attribute.

Parent elements
DataSet

The following is the XML Schema definition of the **DataSet.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.325.2 DataSet.AccentSensitivity

The **DataSet.AccentSensitivity** element indicates whether data in a [DataSet](#) is accent-sensitive. This element is optional. If this element is present, its value MUST be one of the following:

**Auto** (default): The accent-sensitivity setting is auto-derived by querying the **data provider**. If the data provider does not support auto-derivation, this value is interpreted as false.

**True**: Data in the dataset is accent-sensitive.

**False**: Data in the dataset is accent-insensitive.

If this element is not present, its value is interpreted as "Auto".

Following is the parent element for the **DataSet.AccentSensitivity** element.

Parent elements
DataSet

The following is the XML Schema definition of the **DataSet.AccentSensitivity** element.

```
<xsd:element name="AccentSensitivity" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="True" />
      <xsd:enumeration value="False" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.325.3 DataSet.CaseSensitivity

The **DataSet.CaseSensitivity** element specifies whether data in a [DataSet](#) is case-sensitive. This element is optional. If this element is present, its value MUST be one of the following:

**Auto** (default): The case-sensitivity setting is autoderived by querying the **data provider**. If the data provider does not support autoderivation, the value is interpreted as false.

**True**: Data in the dataset is case-sensitive.

**False**: Data in the dataset is case-insensitive.

If the **DataSet.CaseSensitivity** element is not present, its value is interpreted as "Auto".

Following is the parent element of the **DataSet.CaseSensitivity** element.

Parent elements
DataSet

The following is the XML Schema definition of the **DataSet.CaseSensitivity** element.

```
<xsd:element name="CaseSensitivity" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="True" />
      <xsd:enumeration value="False" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.325.4 DataSet.Collation

The **DataSet.Collation** element specifies the locale to use for the collation sequence for sorting data. This element is optional. This element MUST NOT be specified if **DataSet.CollationCulture** is present.

If this element is not present or has an unsupported or invalid value and **DataSet.CollationCulture** is not present, the collation setting is autoderived by querying the **data provider**. If the data provider does not support autoderivation or returns an unsupported or invalid value, the collation that corresponds to the value of the [Report.Language](#) element MUST be used.

The supported locales for this element are as follows:

**Albanian:** The Albanian locale.

**Albanian\_100:** The Albanian locale. [<64>](#)

**Amharic\_100:** The Amharic locale. [<65>](#)

**Arabic:** The Arabic locale.

**Arabic\_100:** The Arabic locale. [<66>](#)

**Armenian\_100:** The Armenian locale. [<67>](#)

**Assamese\_100:** The Assamese locale. [<68>](#)

**Azeri\_Cyrillic\_90:** The Azeri Cyrillic locale. [<69>](#)

**Azeri\_Cyrillic\_100:** The Azeri Cyrillic locale. [<70>](#)

**Azeri\_Latin\_90:** The Azeri Latin locale. [<71>](#)

**Azeri\_Latin\_100:** The Azeri Latin locale. [<72>](#)

**Bashkir\_100:** The Bashkir locale. [<73>](#)

**Bengali\_100:** The Bengali locale. [<74>](#)

**Bosnian\_Cyrillic\_100:** The Bosnian Cyrillic locale. [<75>](#)

**Bosnian\_Latin\_100:** The Bosnian Latin locale. [<76>](#)

**Breton\_100:** The Breton locale. [<77>](#)

**Chinese\_Hong\_Kong\_Stroke\_90:** The Chinese Hong Kong Stroke locale. [<78>](#)

**Chinese\_Hong\_Kong\_Stroke\_100:** The Chinese Hong Kong Stroke locale. [<79>](#)

**Chinese\_Macao\_100:** The Chinese Macao locale. [<80>](#)

**Chinese\_Macao\_Stroke\_100:** The Chinese Macao Stroke locale. [<81>](#)

**Chinese\_PRC:** The Chinese PRC locale.

**Chinese\_PRC\_100:** The Chinese PRC locale. [<82>](#)

**Chinese\_PRC\_Stroke:** The Chinese PRC Stroke locale.

**Chinese\_PRC\_Stroke\_100:** The Chinese PRC Stroke locale. [<83>](#)

**Chinese\_Simplified\_Pinyin\_100:** The Chinese Simplified Pinyin locale. [<84>](#)

**Chinese\_Simplified\_Stroke\_Order\_100:** The Chinese Simplified Stroke locale. [<85>](#)

**Chinese\_Taiwan\_Bopomofo:** The Chinese Taiwan Bopomofo locale.

**Chinese\_Taiwan\_Bopomofo\_100:** The Chinese Taiwan Bopomofo locale. [<86>](#)

**Chinese\_Taiwan\_Stroke**: The Chinese Taiwan Stroke locale.

**Chinese\_Taiwan\_Stroke\_100**: The Chinese Taiwan Stroke locale. [<87>](#)

**Chinese\_Traditional\_Bopomofo\_100**: The Chinese Traditional Bopomofo locale. [<88>](#)

**Chinese\_Traditional\_Pinyin\_100**: The Chinese Traditional Pinyin locale. [<89>](#)

**Chinese\_Traditional\_Stroke\_Count\_100**: The Chinese Traditional Stroke Count locale. [<90>](#)

**Chinese\_Traditional\_Stroke\_Order\_100**: The Chinese Traditional Stroke Order locale. [<91>](#)

**Corsican\_100**: The Corsican locale. [<92>](#)

**Croatian**: The Croatian locale.

**Croatian\_100**: The Croatian locale. [<93>](#)

**Cyrillic\_General**: The Cyrillic General locale.

**Cyrillic\_General\_100**: The Cyrillic General. [<94>](#)

**Czech**: The Czech locale.

**Czech\_100**: The Czech locale. [<95>](#)

**Danish\_Greenlandic\_100**: The Danish Greenlandic locale. [<96>](#)

**Danish\_Norwegian**: The Danish Norwegian locale.

**Dari\_100**: The Dari locale. [<97>](#)

**Divehi\_90**: The Divehi locale. [<98>](#)

**Divehi\_100**: The Divehi locale. [<99>](#)

**Estonian**: The Estonian locale.

**Estonian\_100**: The Estonian locale. [<100>](#)

**Finnish\_Swedish**: The Finnish Swedish locale.

**Finnish\_Swedish\_100**: The Finnish Swedish locale. [<101>](#)

**French**: The French locale.

**French\_100**: The French locale. [<102>](#)

**Frisian\_100**: The Frisian locale. [<103>](#)

**Georgian\_Modern\_Sort**: The Georgian Modern Sort locale.

**Georgian\_Traditional\_100**: The Georgian Traditional locale. [<104>](#)

**German\_PhoneBook**: The German Phonebook locale.

**German\_PhoneBook\_100**: The German Phonebook locale. [<105>](#)

**Greek**: The Greek locale.

**Greek\_100**: The Greek locale. [<106>](#)

**Hebrew**: The Hebrew locale.



**Hebrew\_100:** The Hebrew locale. [<107>](#)

**Hindi:** The Hindi locale.

**Hungarian:** The Hungarian locale.

**Hungarian\_100:** The Hungarian locale. [<108>](#)

**Hungarian\_Technical:** The Hungarian Technical locale.

**Hungarian\_Technical\_100:** The Hungarian Technical locale. [<109>](#)

**Icelandic:** The Icelandic locale.

**Icelandic\_100:** The Icelandic locale. [<110>](#)

**Indic\_General\_90:** The Indic General locale. [<111>](#)

**Indic\_General\_100:** The Indic General locale. [<112>](#)

**Inuktitut\_100:** The Inuktitut locale. [<113>](#)

**Japanese:** The Japanese locale.

**Japanese\_90:** The Japanese locale. [<114>](#)

**Japanese\_100:** The Japanese locale. [<115>](#)

**Japanese\_Bushu\_Kakusu\_100:** The Japanese Bushu Kakusu locale. [<116>](#)

**Japanese\_Radical\_Stroke\_100:** The Japanese Radical Stroke locale. [<117>](#)

**Japanese\_Unicode:** The Japanese Unicode locale. [<118>](#)

**Kazakh\_90:** The Kazakh locale. [<119>](#)

**Kazakh\_100:** The Kazakh locale. [<120>](#)

**Khmer\_100:** The Khmer locale. [<121>](#)

**Korean:** The Korean locale.

**Korean\_90:** The Korean locale. [<122>](#)

**Korean\_100:** The Korean locale. [<123>](#)

**Korean\_Wansung:** The Korean Wansung locale.

**Korean\_Wansung\_Unicode:** The Korean Wansung Unicode locale.

**Lao\_100:** The Lao locale. [<124>](#)

**Latin1\_General:** The Latin1 General locale.

**Latin1\_General\_100:** The Latin1 General locale. [<125>](#)

**Latvian:** The Latvian locale.

**Latvian\_100:** The Latvian locale. [<126>](#)

**Lithuanian:** The Lithuanian locale.

**Lithuanian\_100:** The Lithuanian locale. [<127>](#)

**Lithuanian\_Classic**: The Lithuanian Classic locale. [.<128>](#)

**Macedonian**: The Macedonian locale.

**Macedonian\_FYROM\_90**: The Macedonian FYROM locale. [.<129>](#)

**Macedonian\_FYROM\_100**: The Macedonian FYROM locale. [.<130>](#)

**Maltese\_100**: The Maltese locale. [.<131>](#)

**Maori\_100**: The Maori locale. [.<132>](#)

**Mapudungan\_100**: The Mapudungan locale. [.<133>](#)

**Modern\_Spanish**: The Modern Spanish locale.

**Modern\_Spanish\_100**: The Modern Spanish locale. [.<134>](#)

**Mohawk\_100**: The Mohawk locale. [.<135>](#)

**Mongolian\_100**: The Mongolian locale. [.<136>](#)

**Nepali\_100**: The Nepali locale. [.<137>](#)

**Norwegian\_100**: The Norwegian locale. [.<138>](#)

**Norwegian\_Sami\_100**: The Norwegian Sami locale. [.<139>](#)

**Pashto\_100**: The Pashto locale. [.<140>](#)

**Persian\_100**: The Persian locale. [.<141>](#)

**Polish**: The Polish locale.

**Polish\_100**: The Polish locale. [.<142>](#)

**Romanian**: The Romanian locale.

**Romanian\_100**: The Romanian locale. [.<143>](#)

**Romansh\_100**: The Romansh locale. [.<144>](#)

**Sami\_Norway\_100**: The Sami Norway locale. [.<145>](#)

**Sami\_Sweden\_Finland\_100**: The Sami Sweden Finland locale. [.<146>](#)

**Serbian\_Cyrillic\_100**: The Serbian Cyrillic locale. [.<147>](#)

**Serbian\_Latin\_100**: The Serbian Latin locale. [.<148>](#)

**Slovak**: The Slovak locale.

**Slovak\_100**: The Slovak locale. [.<149>](#)

**Slovenian**: The Slovenian locale.

**Slovenian\_100**: The Slovenian locale. [.<150>](#)

**Swedish\_Finnish\_Sami\_100**: The Swedish Finnish Sami locale. [.<151>](#)

**Syriac\_90**: The Syriac locale. [.<152>](#)

**Syriac\_100**: The Syriac locale. [.<153>](#)

**Tatar\_90**: The Tatar locale. [.<154>](#)

**Tatar\_100**: The Tatar locale. [.<155>](#)

**Tamazight\_100**: The Tamazight locale. [.<156>](#)

**Thai**: The Thai locale.

**Thai\_100**: The Thai locale. [.<157>](#)

**Tibetan\_PRC\_100**: The Tibetan PRC locale. [.<158>](#)

**Traditional\_Spanish**: The Traditional Spanish locale.

**Traditional\_Spanish\_100**: The Traditional Spanish locale. [.<159>](#)

**Turkish**: The Turkish locale.

**Turkish\_100**: The Turkish locale. [.<160>](#)

**Turkmen\_100**: The Turkmen locale. [.<161>](#)

**Uighur\_PRC\_100**: The Uighur PRC locale. [.<162>](#)

**Ukrainian**: The Ukrainian locale.

**Ukrainian\_100**: The Ukrainian locale. [.<163>](#)

**Upper\_Sorbian\_100**: The Upper Sorbian locale. [.<164>](#)

**Urdu\_100**: The Urdu locale. [.<165>](#)

**Uzbek\_Latin\_90**: The Uzbek Latin locale. [.<166>](#)

**Uzbek\_Latin\_100**: The Uzbek Latin locale. [.<167>](#)

**Vietnamese**: The Vietnamese locale.

**Vietnamese\_100**: The Vietnamese locale. [.<168>](#)

**Welsh\_100**: The Welsh locale. [.<169>](#)

**Yakut\_100**: The Yakut locale. [.<170>](#)

**Yi\_100**: The Yi locale. [.<171>](#)

Following is the parent element of the **DataSet.Collation** element.

Parent elements
<a href="#">DataSet</a>

The following is the XML Schema definition of the **DataSet.Collation** element.

```
<xsd:element name="Collation" type="xsd:string" minOccurs="0" />
```

### 2.325.5 DataSet.Fields

The **DataSet.Fields** element specifies a set of fields to include in a [DataSet](#). This element is optional and is of type [Fields](#).

Following is the parent element of the **DataSet.Fields** element.

Parent elements
DataSet

The following is the XML Schema definition of the **DataSet.Fields** element.

```
<xsd:element name="Fields" type="FieldsType" minOccurs="0" />
```

### 2.325.6 DataSet.Filters

The **DataSet.Filters** element specifies a set of filters to apply to each row in a [DataSet](#). This element is optional and is of type [Filters](#).

Following is the parent element of the **DataSet.Filters** element.

Parent elements
DataSet

The following is the XML Schema definition of the **DataSet.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

### 2.325.7 DataSet.InterpretSubtotalsAsDetails

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **DataSet.InterpretSubtotalsAsDetails** element indicates whether rows that represent subtotals that are returned by a data provider that supports **server subtotals** is interpreted as detail rows instead. This element is optional.

If this element is present, its value MUST be one of the following:

**Auto** (default): Subtotal rows are treated as details if the [Report](#) does not use the [Aggregate](#) function to access any fields in the [DataSet](#). Otherwise, subtotal rows SHOULD be treated as subtotal rows.

**True**: Subtotal rows is interpreted as detail rows.

**False**: Subtotal rows are treated as subtotal rows and can be retrieved only by using the **Aggregate** function.

Following is the parent element of the **DataSet.InterpretSubtotalsAsDetails** element.

Parent elements
DataSet

The following is the XML Schema definition of the **DataSet.InterpretSubtotalsAsDetails** element.

```
<xsd:element name="InterpretSubtotalsAsDetails" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="True" />
      <xsd:enumeration value="False" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.325.8 DataSet.KanatypeSensitivity

The **DataSet.KanatypeSensitivity** element specifies whether data in a [DataSet](#) is kanatype-sensitive. This element is optional. If this element is present, its value **MUST** be one of the following:

**Auto** (default): The kanatype-sensitivity setting is autoderived by querying the data provider. If the data provider does not support autoderivation, the value is interpreted as false.

**True**: Data in the dataset is kanatype-sensitive.

**False**: Data in the dataset is kanatype-insensitive.

If this element is not present, its value is interpreted as "Auto".

Following is the parent element of the **DataSet.KanatypeSensitivity** element.

Parent elements
<a href="#">DataSet</a>

The following is the XML Schema definition of the **DataSet.KanatypeSensitivity** element.

```
<xsd:element name="KanatypeSensitivity" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="True" />
      <xsd:enumeration value="False" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.325.9 DataSet.Query

The **DataSet.Query** element specifies the **query** information that is necessary to retrieve data from a data source. This element is optional, and is of type [Query](#). If this element is present the [DataSet.SharedDataSet](#) element **MUST** be omitted.

Following is the parent element of the **DataSet.Query** element.

Parent elements
<a href="#">DataSet</a>

The following is the XML Schema definition of the **DataSet.Query** element.

```
<xsd:element name="Query" type="QueryType" minOccurs="0"/>
```

### 2.325.10 DataSet.SharedDataSet

The **DataSet.SharedDataSet** element specifies the information that is necessary to retrieve data for the report dataset.

This element is optional, and is of type [SharedDataSet](#). If this element is present, the **DataSetQuery** element MUST be omitted.

Following is the parent element of the **DataSet.SharedDataSet** element.

Parent element
<a href="#">DataSet</a>

The following is the XML Schema definition of the **DataSet.SharedDataSet** element

```
<xsd:element name="SharedDataSet" type="SharedDataSetType" minOccurs="0"/>
```

### 2.325.11 DataSet.WidthSensitivity

The **DataSet.WidthSensitivity** element specifies whether data in a [DataSet](#) is width-sensitive. This element is optional. If this element is present, its value MUST be one of the following:

**Auto** (default): The width-sensitivity setting is autoderived by querying the data provider. If the data provider does not support autoderivation, this value is interpreted as false.

**True**: Data in the dataset is width-sensitive.

**False**: Data in the dataset is width-insensitive.

If this element is not present, its value is interpreted as "Auto".

Following is the parent element of the **DataSet.WidthSensitivity** element.

Parent elements
DataSet

The following is the XML Schema definition of the **DataSet.WidthSensitivity** element.

```
<xsd:element name="WidthSensitivity" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="True" />
      <xsd:enumeration value="False" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

## 2.325.12 DataSet.DefaultRelationships

*Applies to [RDL 2011/01](#)*

The **DataSet.DefaultRelationships** element specifies a collection of default **relationships** to use between the containing [DataSet](#) and the **related dataset** when nested scopes reference different **DataSets**. The **DataSet.DefaultRelationships** element is optional and MUST NOT be specified more than once. If this element is specified, it is of type [DefaultRelationships](#).

Following is the parent element of the **DataSet.DefaultRelationships** element.

Parent elements
<a href="#">DataSet</a>

The following is the XML Schema definition of the **DataSet.DefaultRelationships** element.

```
<xsd:element name="DefaultRelationships" type="DefaultRelationshipsType" minOccurs="0" />
```

## 2.325.13 DataSet.NullsAsBlanks

*Applies to [RDL 2011/01](#)*

The **DataSet.NullsAsBlanks** element specifies whether data in a [DataSet](#) uses blank semantics. The **DataSet.NullsAsBlanks** element is optional. If this element is present, its value MUST be a [Boolean](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.2). If the **DataSet.NullsAsBlanks** element is not present, its value is interpreted as false. [<172>](#)

Following is the parent element of the **DataSet.NullsAsBlanks** element.

Parent elements
<a href="#">DataSet</a>

The following is the XML Schema definition of the **DataSet.NullsAsBlanks** element.

```
<xsd:element name="NullsAsBlanks" type="xsd:boolean" minOccurs="0" />
```

## 2.325.14 DataSet.CollationCulture

*Applies to [RDL 2011/01](#)*

The **DataSet.CollationCulture** element specifies the locale to use for the collation sequence for sorting data. The **DataSet.CollationCulture** element is optional. If this element is present, its value MUST be a [ReportLanguage](#). If this element is present, it MUST NOT be an expression. The **DataSet.CollationCulture** element MUST NOT be specified if [DataSet.Collation](#) is present. If the **DataSet.CollationCulture** element is not present, the collation sequence is determined by the **DataSet.Collation** element.

Following is the parent element of the **DataSet.CollationCulture** element.

Parent elements
<a href="#">DataSet</a>

The following is the XML Schema definition of the **DataSet.CollationCulture** element.

```
<xsd:element name="CollationCulture" type="xsd:string" minOccurs="0" />
```

## 2.326 DataSet

The **DataSet** element specifies information that is necessary to retrieve data for a DataSet.

The following are the parent and child elements of the **DataSet** element.

Parent element
<a href="#">DataSet</a>

Child elements
<a href="#">DataSet.SharedDataSetReference</a>
<a href="#">DataSet.QueryParameters</a>

The following is the XML Schema definition of the **DataSet** element.

```
<xsd:complexType name="DataSetType">  
  <xsd:choice minOccurs="1" maxOccurs="unbounded">  
    <xsd:element name="SharedDataSetReference" type="xsd:string" />  
    <xsd:element name="QueryParameters" type="QueryParametersType" minOccurs="0" />  
    <xsd:any namespace="##other" processContents="skip" />  
  </xsd:choice>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

### 2.326.1 DataSet.SharedDataSetReference

The **DataSet.SharedDataSetReference** element specifies the absolute path (such as "/datasets/sales") or the relative path (such as "sales") to a shared dataset on the current report server. Relative paths start in the same folder as the referencing report.

This element MUST be specified. The value of this element MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1).

The following is the parent element of the **DataSet.SharedDataSetReference** element.

Parent element
<a href="#">DataSet</a>

The following is the XML Schema definition of the **DataSet.SharedDataSetReference** element.

```
<xsd:element name="SharedDataSetReference" type="xsd:string" />
```



## 2.326.2 SharedDataSet.QueryParameters

The **SharedDataSet.QueryParameters** element specifies a list of query parameters that are passed to the data source as part of the query. This element is optional and is of type [QueryParameters](#).

The following is the parent element of the **SharedDataSet.QueryParameters** element.

Parent element
<a href="#">SharedDataSet</a>

The following is the XML Schema definition of the **SharedDataSet.QueryParameters** element.

```
<xsd:element name="QueryParameters" type="QueryParametersType" minOccurs="0" />
```

## 2.327 Fields

The **Fields** element specifies the collection of [Field](#) elements that define the fields in a [DataSet](#).

The following are the parent and child elements of the **Fields** element.

Parent elements
DataSet

Child elements
<a href="#">Fields.Field</a>

The following is the XML Schema definition of the **Fields** element.

```
<xsd:complexType name="FieldsType">  
  <xsd:sequence>  
    <xsd:element name="Field" type="FieldType" maxOccurs="unbounded" />  
  </xsd:sequence>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

### 2.327.1 Fields.Field

The **Fields.Field** element specifies the [Field](#) elements within the [Fields](#) collection. The **Fields.Field** element is of type **Field**.

Following is the parent element of the **Fields.Field** element.

Parent elements
Fields

The following is the XML Schema definition of the **Fields.Field** element.

```
<xsd:element name="Field" type="FieldType" maxOccurs="unbounded" />
```

## 2.328 Field

The **Field** element specifies information about a field. This element MUST specify either the **DataField** or **Value** child elements.

The following are the parent elements, attributes, and child elements of the **Field** element.

Parent elements
<a href="#">Fields</a>

Attributes
<a href="#">Field.Name</a>

Child elements
<a href="#">Field.DataField</a>
<a href="#">Field.Value</a>

**Applies to** [RDL 2011/01](#)

Child elements
<a href="#">Field.AggregateIndicatorField</a>

The following is the XML Schema definition of the **Field** element in [RDL 2003/10](#) and [RDL 2005/01](#).

```
<xsd:complexType name="FieldType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataField" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **Field** element in [RDL 2008/01](#).

```
<xsd:complexType name="FieldType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataField" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="StringWithDataModelAttribute" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **Field** element in [RDL 2010/01](#) and [RDL 2016/01](#).

**Note** The following XSD represents RDL macro-versioned schemas only. Possible additions, identified earlier in this section, to base schema RDL 2010/01 from micro-versioned schemas RDL 2011/01, RDL 2012/01, and RDL 2013/01 are provided in sections 5.5, [5.6](#), and [5.7](#), respectively. For more information about macro- and micro-versioned schemas, see section [2.1](#).

```
<xsd:complexType name="FieldType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataField" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="StringWithDataTypeAttribute" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.328.1 Field.Name

The **Field.Name** attribute specifies a unique identifier for a [Field](#). This attribute MUST be specified. The value of the **Field.Name** attribute MUST be a case-sensitive **CLS-compliant identifier** [\[UTR15\]](#) that is unique among the fields within the same [Fields](#) collection.

Following is the parent element of the **Field.Name** attribute.

Parent elements
Field

The following is the XML Schema definition of the **Field.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

### 2.328.2 Field.DataField

The **Field.DataField** element specifies the name of the field that is returned by the query. Data field names do not have to be unique. Multiple [Field](#) elements can refer to the same data field name.

This element is optional. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1), and the [Field.Value](#) element MUST NOT be specified.

Following is the parent element of the **Field.DataField** element.

Parent elements
Field

The following is the XML Schema definition of the **Field.DataField** element.

```
<xsd:element name="DataField" type="xsd:string" minOccurs="0" />
```

### 2.328.3 Field.Value

The **Field.Value** element specifies a value or an expression that evaluates to a value for this [Field](#). The expression MUST NOT contain **aggregate functions** or references to report items.

The **Field.Value** element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1), and the [Field.DataField](#) element MUST NOT be specified.

The following are the parent element and attribute of the **Field.Value** element.

Parent elements
Field

Attributes
<a href="#">Field.Value.DataType</a>

The following is the XML Schema definition of the **Field.Value** element in [RDL 2003/10](#) and [RDL 2005/01](#).

```
<xsd:element name="Value" type="xsd:string" minOccurs="0" />
```

The following is the XML Schema definition of the **Field.Value** element in [RDL 2008/01](#), [RDL 2010/01](#) and [RDL 2016/01](#).

```
<xsd:element name="Value" type="StringWithDataModelAttribute" minOccurs="0" />
```

#### 2.328.4 **Field.Value.DataType**

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **Field.Value.DataType** attribute specifies the data type of a value if the value is a constant. This attribute is optional. If this attribute is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1). The **String** value MUST be the name of one of the following RDL data types: **String**, [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2), [DateTime](#) ([XMLSCHEMA2] section 3.2.7), [Integer](#) ([XMLSCHEMA2/2] section 3.3.17), or [Float](#) ([XMLSCHEMA2] section 3.2.4). If this attribute is not present, the value of the [Field.Value](#) element is interpreted as a **String**.

Following is the parent element of the **Field.Value.DataType** attribute.

Parent elements
Field.Value

#### 2.328.5 **Field.AggregateIndicatorField**

**Applies to** [RDL 2011/01](#)

The **Field.AggregateIndicatorField** element specifies the name of another [Field](#) that specifies whether the data is aggregated over the current **Field**. The referenced **Field** MUST evaluate to a [Boolean](#) value. If the value evaluates to true, the data in the current row is aggregated over this field. If the value evaluates to false the data in the current row is grouped by this field.

This element is optional. If this element is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that contains the name of another **Field** in the same [DataSet](#).<173>

Following is the parent element of the **Field.AggregateIndicatorField** element.

Parent elements
Field

The following is the XML Schema definition of the **Field.AggregateIndicatorField** element.

```
<xsd:element name="AggregateIndicatorField" type="xsd:string" minOccurs="0" />
```

## 2.329 Filters

The **Filters** element specifies an ordered list of [Filter](#) elements that apply to a [DataSet](#) (in [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)) or a group.

The following are the parent and child elements of the **Filters** element.

Parent elements
<a href="#">Tablix</a>
<a href="#">List</a>
<a href="#">Matrix</a>
<a href="#">Table</a>
<a href="#">Group</a>
<a href="#">Grouping</a>
<a href="#">Chart</a>
<a href="#">GaugePanel</a>
<a href="#">CustomData</a>
DataSet

Child elements
<a href="#">Filters.Filter</a>

The following is the XML Schema definition of the **Filters** element.

```
<xsd:complexType name="FiltersType">  
  <xsd:sequence>  
    <xsd:element name="Filter" type="FilterType" maxOccurs="unbounded" />  
  </xsd:sequence>  
  <xsd:anyAttribute namespace="##other" processContents="skip" />  
</xsd:complexType>
```

### 2.329.1 Filters.Filter

The **Filters.Filter** element specifies a [Filter](#) that is used to restrict the rows within a [DataSet](#) (in [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)), a group, or a data region. **Filters.Filter** elements are

applied in the order in which they are specified within the parent [Filters](#) collection. The **Filters.Filter** element is of type **Filter**.

Following is the parent element of the **Filters.Filter** element.

Parent elements
<a href="#">Filters</a>

The following is the XML Schema definition of the **Filters.Filter** element.

```
<xsd:element name="Filter" type="FilterType" maxOccurs="unbounded" />
```

### 2.330 Filter

The **Filter** element specifies a **filter** to apply to rows of data within a [DataSet](#) (in [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)), a group, or a data region.

The following are the parent and child elements of the **Filter** element.

Parent elements
<a href="#">Filters</a>

Child elements
<a href="#">Filter.FilterExpression</a>
<a href="#">Filter.FilterValues</a>
<a href="#">Filter.Operator</a>

The following is the XML Schema definition of the **Filter** element.

```
<xsd:complexType name="FilterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="FilterExpression" type="xsd:string" />
    <xsd:element name="Operator">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Equal" />
          <xsd:enumeration value="Like" />
          <xsd:enumeration value="NotEqual" />
          <xsd:enumeration value="GreaterThan" />
          <xsd:enumeration value="GreaterThanOrEqual" />
          <xsd:enumeration value="LessThan" />
          <xsd:enumeration value="LessThanOrEqual" />
          <xsd:enumeration value="TopN" />
          <xsd:enumeration value="BottomN" />
          <xsd:enumeration value="TopPercent" />
          <xsd:enumeration value="BottomPercent" />
          <xsd:enumeration value="In" />
          <xsd:enumeration value="Between" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="FilterValues" type="FilterValuesType" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
</xsd:complexType>
```

```

</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.330.1 Filter.FilterExpression

The **Filter.FilterExpression** element specifies an expression that is evaluated for each instance of a group, or for each row of data that is associated with a [DataSet](#) (in [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)), a group, or a data region. This expression is then compared to the value of the [Filter.FilterValues](#) element by using the [Filter.Operator](#) element. Failed comparisons MUST result in the row or instance being filtered out of its containing group, dataset, or data region.

The **Filter.FilterExpression** element MUST be specified, and it MUST NOT contain references to report items. If the **Filter.FilterExpression** element is not within a group, its value MUST NOT contain an **aggregate function**. If this element is within a group, it MUST NOT contain any instances of the [RunningValue](#) aggregate function, and it MUST NOT contain any [First](#) or [Last](#) aggregate functions with anything but the default (current) scope.

Following is the parent element of the **Filter.FilterExpression** element.

Parent elements
<a href="#">Filter</a>

The following is the XML Schema definition of the **Filter.FilterExpression** element.

```
<xsd:element name="FilterExpression" type="xsd:string" />
```

### 2.330.2 Filter.FilterValues

The **Filter.FilterValues** element specifies the values to compare to a [Filter.FilterExpression](#). The **Filter.FilterValues** element MUST be specified and is of type [FilterValues](#).

Following is the parent element of the **Filter.FilterValues** element.

Parent elements
<a href="#">Filter</a>

The following is the XML Schema definition of the **Filter.FilterValues** element.

```
<xsd:element name="FilterValues" type="FilterValuesType" />
```

### 2.330.3 Filter.Operator

The **Filter.Operator** element specifies an operator to use to compare the values of [Filter.FilterExpression](#) and [FilterValues](#). The **Filter.Operator** element MUST be specified. The value of this element MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is one of the following:

**Equal:** Specifies an equality comparison.

**Like:** Specifies a like comparison (that is, it compares a string against a pattern).

**NotEqual:** Specifies an inequality comparison.

**GreaterThan:** Specifies a comparison to determine whether the **Filter.FilterExpression** value is greater than the **FilterValues** values.

**GreaterThanOrEqual:** Specifies a comparison to determine whether the **Filter.FilterExpression** value is greater than or equal to the **FilterValues** values.

**LessThan:** Specifies a comparison to determine whether the **Filter.FilterExpression** value is less than the **FilterValues** values.

**LessThanOrEqual:** Specifies a comparison to determine whether the **Filter.FilterExpression** value is less than or equal to the **FilterValues** values.

**TopN:** Specifies whether the **Filter.FilterExpression** value is within the top *N* data values, where *N* is specified by [FilterValues.FilterValue](#).

**BottomN:** Specifies whether the **Filter.FilterExpression** value is within the bottom *N* data values, where *N* is specified by **FilterValues.FilterValue**.

**TopPercent:** Specifies whether the **Filter.FilterExpression** value is within the top *N* percent of data values, where *N* is specified by **FilterValues.FilterValue**.

**BottomPercent:** Specifies whether the **Filter.FilterExpression** value is within the bottom *N* percent of data values, where *N* is specified by **FilterValues.FilterValue**.

**In:** Specifies whether the value of **Filter.FilterExpression** is equal to any **FilterValues.FilterValue** instances.

**Between:** Specifies whether the value of **Filter.FilterExpression** is between the values of two **FilterValues.FilterValue** instances.

The **TopN**, **BottomN**, **TopPercent**, and **BottomPercent** operators SHOULD include ties in the resulting data. String comparisons within filters SHOULD be locale-dependent. The NULL constant is equal to itself.

The **TopPercent** and **BottomPercent** operators SHOULD round up and down, respectively, if the percentage that is specified would result in a partial item being included (for example, if the Top 25% of 13 items is four items and the Bottom 75% is nine items). If the value of the **Filter.Operator** element is "In", multi-value report parameters that have the [ReportParameter.MultiValue](#) element specified with a value of true MUST be supported and treated as multiple **FilterValue** instances.

In the following example, if "Cities" is a multi-value parameter for a [Filter](#) that references "Cities" within any **FilterValues.FilterValue** elements, the following RDL:

```
<FilterValues>
  <FilterValue>=Parameters!Cities.Value</FilterValue>
</FilterValues>
```

MUST be equivalent to the following RDL:

```
<FilterValues>
  <FilterValue>=Parameters!Cities.Value[0]</FilterValue>
  <FilterValue>=Parameters!Cities.Value[1]</FilterValue>
  [...]
</FilterValues>
```

Following is the parent element of the **Filter.Operator** element.



<b>Parent elements</b>
Filter

The following is the XML Schema definition of the **Filter.Operator** element.

```
<xsd:element name="Operator">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Equal" />
      <xsd:enumeration value="Like" />
      <xsd:enumeration value="NotEqual" />
      <xsd:enumeration value="GreaterThan" />
      <xsd:enumeration value="GreaterThanOrEqual" />
      <xsd:enumeration value="LessThan" />
      <xsd:enumeration value="LessThanOrEqual" />
      <xsd:enumeration value="TopN" />
      <xsd:enumeration value="BottomN" />
      <xsd:enumeration value="TopPercent" />
      <xsd:enumeration value="BottomPercent" />
      <xsd:enumeration value="In" />
      <xsd:enumeration value="Between" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.331 FilterValues

The **FilterValues** element specifies an ordered list of **FilterValue** elements to compare against the value of the [Filter.FilterExpression](#) element within the same [Filter](#).

The following are the parent and child elements of the **FilterValues** element.

<b>Parent elements</b>
Filter

<b>Child elements</b>
<a href="#">FilterValues.FilterValue</a>

The following is the XML Schema definition of the **FilterValues** element in [RDL 2003/10](#).

```
<xsd:complexType name="FilterValuesType">
  <xsd:sequence>
    <xsd:element name="FilterValue" type="xsd:string"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
```

The following is the XML Schema definition of the **FilterValues** element in [RDL 2005/01](#).

```
<xsd:complexType name="FilterValuesType">
  <xsd:sequence>
    <xsd:element name="FilterValue" type="xsd:string"
      maxOccurs="unbounded" />
  </xsd:sequence>
```

```

    <xsd:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>

```

The following is the XML Schema definition of the **FilterValues** element in [RDL 2008/01](#).

```

<xsd:complexType name="FilterValuesType">
  <xsd:sequence>
    <xsd:element name="FilterValue" type="StringWithDataAttribute"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

The following is the XML Schema definition of the **FilterValues** element in [RDL 2010/01](#) and [RDL 2016/01](#).

```

<xsd:complexType name="FilterValuesType">
  <xsd:sequence>
    <xsd:element name="FilterValue" type="StringWithDataAttribute"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

### 2.331.1 FilterValues.FilterValue

The **FilterValues.FilterValue** element specifies a value to compare to the value of the [Filter.FilterExpression](#) element by using the [Filter.Operator](#) element.

If **Filter.Operator** is not set to "In" or "Between", there MUST be exactly one **FilterValues.FilterValue** element.

If **Filter.Operator** is set to "TopN" or "BottomN", the value of the **FilterValues.FilterValue** element MUST be an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) or an expression that evaluates to an **Integer**.

If **Filter.Operator** is set to "TopPercent" or "BottomPercent", the value of the **FilterValues.FilterValue** element MUST be a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) or an expression that evaluates to a **Float**.

If **Filter.Operator** is set to "Between", there MUST be exactly two **FilterValue** elements.

The following are the parent element and attribute of the **FilterValues.FilterValue** element.

Parent elements
<a href="#">FilterValues</a>

Attributes
<a href="#">FilterValues.FilterValue.DataType</a>

The following is the XML Schema definition of the **FilterValues.FilterValue** element in [RDL 2003/10](#) and [RDL 2005/01](#).

```
<xsd:element name="FilterValue" type="xsd:string"
  maxOccurs="unbounded" />
```

The following is the XML Schema definition of the **FilterValues.FilterValue** element in [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#).

```
<xsd:element name="FilterValue" type="StringWithDataModelAttribute" maxOccurs="unbounded" />
```

### 2.331.2 FilterValues.FilterValue.DataType

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **FilterValues.FilterValue.DataType** attribute specifies the data type of the **FilterValue** if the value of the attribute is a constant. The **FilterValues.FilterValue.DataType** attribute is optional. If this attribute is present, its value MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1), [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2), [DateTime](#) ([XMLSCHEMA2] section 3.2.7), [Integer](#) ([XMLSCHEMA2/2] section 3.3.17), or [Float](#) ([XMLSCHEMA2] section 3.2.4). If this attribute is not present, its value is interpreted as a **String**.

Following is the parent element of the **FilterValues.FilterValue.DataType** attribute.

Parent elements
<a href="#">FilterValues.FilterValue</a>

### 2.332 Query

The **Query** element specifies the information that is necessary to execute and retrieve data for a [DataSet](#).

The following are the parent and child elements of the **Query** element.

Parent elements
<a href="#">DataSet</a>

Child elements
<a href="#">Query.CommandText</a>
<a href="#">Query.CommandType</a>
<a href="#">Query.DataSourceName</a>
<a href="#">Query.QueryParameters</a>
<a href="#">Query.Timeout</a>

The following is the XML Schema definition of the **Query** element.

```
<xsd:complexType name="QueryType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
```

```

<xsd:element name="DataSourceName" type="xsd:string" />
<xsd:element name="CommandType" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Text" />
      <xsd:enumeration value="StoredProcedure" />
      <xsd:enumeration value="TableDirect" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="CommandText" type="xsd:string" />
<xsd:element name="QueryParameters" type="QueryParametersType"
  minOccurs="0" />
<xsd:element name="Timeout" type="xsd:unsignedInt" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

### 2.332.1 Query.CommandText

The **Query.CommandText** element specifies the **query** to execute to obtain data for a [DataSet](#). This element **MUST** be specified. The value of this element **MUST** be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) or an expression that evaluates to a **String**.

Following is the parent element of the **Query.CommandText** element.

Parent elements
<a href="#">Query</a>

The following is the XML Schema definition of the **Query.CommandText** element.

```

<xsd:element name="CommandText" type="xsd:string" />

```

### 2.332.2 Query.CommandType

The **Query.CommandType** element specifies the type of the query that is contained in the [Query.CommandText](#). The **Query.CommandType** element is optional. If this element is present, its value **MUST** be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is one of the following:

**Text** (default): The **Query.CommandType** element contains a query command to execute.

**StoredProcedure**: The **Query.CommandType** element contains the name of a stored procedure to execute.

**TableDirect**: The **Query.CommandType** element contains the name of a [Table](#) from which to retrieve rows.

If this element is not present, its value is interpreted as "Text". Following is the parent element of the **Query.CommandType** element.

Parent elements
<a href="#">Query</a>

The following is the XML Schema definition of the **Query.CommandType** element.

```

<xsd:element name="CommandType" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Text" />
      <xsd:enumeration value="StoredProcedure" />
      <xsd:enumeration value="TableDirect" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

```

### 2.332.3 Query.DataSourceName

The **Query.DataSourceName** element specifies the name of a data source against which to execute a query. This element MUST be specified. The value of this element MUST be a [String](#) ([XMLSCHEMA2/2] section 3.2.1) that is the same as the name of a [DataSource](#) that is present in [Report.DataSources](#).

Following is the parent element of the **Query.DataSourceName** element.

Parent elements
<a href="#">Query</a>

The following is the XML Schema definition of the **Query.DataSourceName** element.

```

<xsd:element name="DataSourceName" type="xsd:string" />

```

### 2.332.4 Query.QueryParameters

The **Query.QueryParameters** element specifies a list of query parameters that are passed to the data source as part of the query. This element is optional and is of type [QueryParameters](#).

Following is the parent element of the **Query.QueryParameters** element.

Parent elements
<a href="#">Query</a>

The following is the XML Schema definition of the **Query.QueryParameters** element.

```

<xsd:element name="QueryParameters" type="QueryParametersType" minOccurs="0" />

```

### 2.332.5 Query.Timeout

The **Query.Timeout** element specifies the number of seconds to allow the query to execute before the query times out. This element is optional.

If this element is present, its value MUST be an [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) that is greater than or equal to 0 and less than or equal to 2147483647. If this element is not present, its value is interpreted as 0. A value of 0 indicates that the query does not time out.

Following is the parent element of the **Query.Timeout** element.

Parent elements
<a href="#">Query</a>

The following is the XML Schema definition of the **Query.Timeout** element.

```
<xsd:element name="Timeout" type="xsd:unsignedInt" minOccurs="0" />
```

### 2.333 QueryParameters

The **QueryParameters** element specifies an ordered list of [QueryParameter](#) elements whose values are passed to a data source as part of a **query**.

The following are the parent and child elements of the **QueryParameters** element.

Parent elements
<a href="#">Query</a>
<a href="#">SharedDataSet</a>

Child elements
<a href="#">QueryParameters.QueryParameter</a>

The following is the XML Schema definition of the **QueryParameters** element.

```
<xsd:complexType name="QueryParametersType">
  <xsd:sequence>
    <xsd:element name="QueryParameter" type="QueryParameterType"
      minOccurs="0" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

#### 2.333.1 QueryParameters.QueryParameter

The **QueryParameters.QueryParameter** element specifies a [QueryParameter](#) in the collection of [QueryParameters](#) for the current [Query](#). The **QueryParameters.QueryParameter** element is of type **QueryParameter**.

Following is the parent element of the **QueryParameters.QueryParameter** element.

Parent elements
<a href="#">QueryParameters</a>

The following is the XML Schema definition of the **QueryParameters.QueryParameter** element.

```
<xsd:element name="QueryParameter" type="QueryParameterType"
  minOccurs="0" maxOccurs="unbounded" />
```

## 2.334 QueryParameter

The **QueryParameter** element specifies information about a parameter that is passed to a data source as part of a [Query](#).

The following are the parent element, attribute, and child element of the **QueryParameter** element.

Parent elements
<a href="#">QueryParameters</a>

Attributes
<a href="#">QueryParameter.Name</a>

Child elements
<a href="#">QueryParameter.Value</a>

The following is the XML Schema definition of the **QueryParameter** element in [RDL 2003/10](#) and [RDL 2005/01](#).

```
<xsd:complexType name="QueryParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **QueryParameter** element in [RDL 2008/01](#).

```
<xsd:complexType name="QueryParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="StringWithDataModelAttribute" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

The following is the XML Schema definition of the **QueryParameter** element in [RDL 2010/01](#) and [RDL 2016/01](#).

```
<xsd:complexType name="QueryParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="StringWithDataModelAttribute" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.334.1 QueryParameter.Name

The **QueryParameter.Name** attribute specifies a unique identifier for a [QueryParameter](#) instance. The **QueryParameter.Name** attribute MUST be specified. The value of this attribute MUST be a case-sensitive **CLS-compliant identifier** [UTR15] that is unique among the **QueryParameter** elements that belong to the same [QueryParameters](#) element.

Following is the parent element of the **QueryParameter.Name** attribute.

Parent elements
<a href="#">QueryParameter</a>

The following is the XML Schema definition of the **QueryParameter.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:string" use="required" />
```

### 2.334.2 QueryParameter.Value

The **QueryParameter.Value** element specifies an expression that evaluates to the value to pass to a data source when executing the current [Query](#). The expression can refer to [ReportParameters](#), but it MUST NOT contain references to report items or fields, and it MUST NOT use aggregate functions.

When a query parameter is used in a query to retrieve the data for **ReportParameter.Values** or [ReportParameter.DefaultValue](#) and when the expression contains a [ReportParameter](#) reference, the expression MUST reference only **ReportParameters** that occur earlier in the ordered list of **ReportParameters** than the referencing **ReportParameter**. The value for this [QueryParameter](#) is then taken from the value that is specified or chosen for the earlier **ReportParameter**.

The **QueryParameter.Value** element MUST be specified, and its value MUST be a **Variant** or **VariantArray** or an expression that evaluates to a **Variant** or **VariantArray**.

The following are the parent element and attribute of the **QueryParameter.Value** element.

Parent elements
<a href="#">QueryParameter</a>

Attributes
<a href="#">QueryParameter.Value.DataType</a>

The following is the XML Schema definition of the **QueryParameter.Value** element in [RDL 2003/10](#) and [RDL 2005/01](#).

```
<xsd:element name="Value" type="xsd:string" minOccurs="0" />
```

The following is the XML Schema definition of the **QueryParameter.Value** element in [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#).

```
<xsd:element name="Value" type="StringWithDataTypeAttribute" minOccurs="0" />
```



### 2.334.3 QueryParameter.Value.DataType

Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

The **QueryParameter.Value.DataType** attribute specifies the data type of a value if the value is a constant. The **QueryParameter.Value.DataType** attribute is optional. If this attribute is present, its value MUST be the name of one of the following RDL data types: [String](#) ([XMLSCHEMA2/2] section 3.2.1), [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2), [DateTime](#) ([XMLSCHEMA2] section 3.2.7), [Integer](#) ([XMLSCHEMA2/2] section 3.3.17), or [Float](#) ([XMLSCHEMA2] section 3.2.4). If this attribute is not present, the constant value is interpreted as a **String**.

Following is the parent element of the **QueryParameter.Value.DataType** attribute.

Parent elements
<a href="#">QueryParameter.Value</a>

### 2.335 DefaultRelationships

Applies to [RDL 2011/01](#)

The **DefaultRelationships** element specifies a collection of [DefaultRelationship](#) elements.

The following are the parent and child elements of the **DefaultRelationships** element.

Parent elements
<a href="#">DataSet</a>

Child elements
<a href="#">DefaultRelationships.DefaultRelationship</a>

The following is the XML Schema definition of the **DefaultRelationships** element.

```
<xsd:complexType name="DefaultRelationshipsType">
  <xsd:sequence>
    <xsd:element name="DefaultRelationship" type="DefaultRelationshipType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

#### 2.335.1 DefaultRelationships.DefaultRelationship

Applies to [RDL 2011/01](#)

The **DefaultRelationships.DefaultRelationship** element specifies a default relationship to another [DataSet](#). The **DefaultRelationships.DefaultRelationship** element MUST be specified at least once and can be specified more than once. This element is of type [DefaultRelationship](#).

Following is the parent element of the **DefaultRelationships.DefaultRelationship** element.

<b>Parent elements</b>
<a href="#">DefaultRelationships</a>

The following is the XML Schema definition of the **DefaultRelationships.DefaultRelationship** element.

```
<xsd:element name="DefaultRelationship" type="DefaultRelationshipType" minOccurs="1"
maxOccurs="unbounded" />
```

## 2.336 DefaultRelationship

**Applies to** [RDL 2011/01](#)

The **DefaultRelationship** element specifies a default **relationship** to use between the containing [DataSet](#) and a [DefaultRelationship.RelatedDataSet](#) when the scope that is bound to the containing **DataSet** is a child scope of the scope bound to the **DefaultRelationship.RelatedDataSet**. [<174><175>](#)

The following are the parent and child elements of the **DefaultRelationship** element.

<b>Parent elements</b>
<a href="#">DefaultRelationships</a>

<b>Child elements</b>
DefaultRelationship.RelatedDataSet
<a href="#">DefaultRelationship.NaturalJoin</a>
<a href="#">DefaultRelationship.JoinConditions</a>

The following is the XML Schema definition of the **DefaultRelationship** element.

```
<xsd:complexType name="DefaultRelationshipType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="RelatedDataSet" type="xsd:string" />
    <xsd:element name="NaturalJoin" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="JoinConditions" type="JoinConditionsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.336.1 DefaultRelationship.RelatedDataSet

**Applies to** [RDL 2011/01](#)

The **DefaultRelationship.RelatedDataSet** element specifies the name of the target [DataSet](#) for this [DefaultRelationship](#). The **DefaultRelationship.RelatedDataSet** element **MUST** be specified and **MUST NOT** be specified more than once. The value of this element **MUST** be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1). [<176>](#)

Following is the parent element of the **DefaultRelationship.RelatedDataSet** element.

<b>Parent elements</b>
------------------------

<a href="#">DefaultRelationship</a>
-------------------------------------

The following is the XML Schema definition of the **DefaultRelationship.RelatedDataSet** element.

```
<xsd:element name="RelatedDataSet" type="xsd:string" />
```

### 2.336.2 DefaultRelationship.JoinConditions

*Applies to* [RDL 2011/01](#)

The **DefaultRelationship.JoinConditions** element specifies the **join** conditions that specify which instances of the containing [DataSet](#) scope correspond to which instances of the [DefaultRelationship.RelatedDataSet](#) scope. The **DefaultRelationship.JoinConditions** element is optional and MUST NOT be specified more than once. This element is of type [JoinConditions](#).<177>

Following is the parent element of the **DefaultRelationship.JoinConditions** element.

<b>Parent elements</b>
------------------------

<a href="#">DefaultRelationship</a>
-------------------------------------

The following is the XML Schema definition of the **DefaultRelationship.JoinConditions** element.

```
<xsd:element name="JoinConditions" type="JoinConditionsType" minOccurs="0" />
```

### 2.336.3 DefaultRelationship.NaturalJoin

*Applies to* [RDL 2011/01](#)

The **DefaultRelationship.NaturalJoin** element specifies that the containing dataset is in the same order as the **RelatedDataSet** with respect to the **join** condition. The **DefaultRelationship.NaturalJoin** element is optional and MUST NOT be specified more than once. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2](#) section 3.2.2). If this element is not present, its value is interpreted as false.<178><179>

Following is the parent element of the **DefaultRelationship.NaturalJoin** element.

<b>Parent elements</b>
------------------------

<a href="#">DefaultRelationship</a>
-------------------------------------

The following is the XML Schema definition of the **DefaultRelationship.NaturalJoin** element.

```
<xsd:element name="NaturalJoin" type="xsd:boolean" minOccurs="0" />
```

### 2.337 JoinConditions

*Applies to* [RDL 2011/01](#)

The **JoinConditions** element specifies a collection of [JoinCondition](#) elements.

The following are the parent and child elements of the **JoinConditions** element.

<b>Parent elements</b>
------------------------

<a href="#">DefaultRelationship</a>
-------------------------------------

<b>Child elements</b>
-----------------------

<a href="#">JoinConditions.JoinCondition</a>
--

The following is the XML Schema definition of the **JoinConditions** element.

```
<xsd:complexType name="JoinConditionsType">
  <xsd:sequence>
    <xsd:element name="JoinCondition" type="JoinConditionType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.337.1 JoinConditions.JoinCondition

*Applies to [RDL 2011/01](#)*

The **JoinConditions.JoinCondition** element specifies a condition to use for the **join**. This element MUST be specified at least once and can be specified more than once. The **JoinConditions.JoinCondition** element is of type [JoinCondition](#).

Following is the parent element of the **JoinConditions.JoinCondition** element.

<b>Parent elements</b>
------------------------

<a href="#">JoinConditions</a>
--------------------------------

The following is the XML Schema definition of the **JoinConditions.JoinCondition** element.

```
<xsd:element name="JoinCondition" type="JoinConditionType" minOccurs="1"
  maxOccurs="unbounded" />
```

### 2.338 JoinCondition

*Applies to [RDL 2011/01](#)*

The **JoinCondition** element specifies a join condition used in a **join**.

The following are the parent and child elements of the **JoinCondition** element.

<b>Parent elements</b>
------------------------

<a href="#">JoinConditions</a>
--------------------------------

<b>Child elements</b>
-----------------------

<a href="#">JoinCondition.ForeignKey</a>
--

<b>Child elements</b>
<a href="#">JoinCondition.PrimaryKey</a>
<a href="#">JoinCondition.SortDirection</a>

The following is the XML Schema definition of the **JoinCondition** element.

```
<xsd:complexType name="JoinConditionType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ForeignKey" type="rdl2010:StringWithDataModelAttribute" />
    <xsd:element name="PrimaryKey" type="rdl2010:StringWithDataModelAttribute" />
    <xsd:element name="SortDirection" minOccurs="0" maxOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Ascending"/>
          <xsd:enumeration value="Descending"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.338.1 JoinCondition.ForeignKey

*Applies to [RDL 2011/01](#)*

The **JoinCondition.ForeignKey** element specifies an expression to evaluate in the containing [DataSet](#). The **JoinCondition.ForeignKey** element MUST be specified and MUST NOT be specified more than once. The value of this element MUST be an expression that evaluates to a **Variant**. The value of this element MUST NOT contain a reference to [Variable](#), **ReportItem**, **RunningValue**, **Previous**, or any aggregate functions. [<180>](#)

<b>Parent elements</b>
<a href="#">JoinCondition</a>

The following is the XML Schema definition of the **JoinCondition.ForeignKey** element.

```
<xsd:element name="ForeignKey" type="xsd:string" />
```

### 2.338.2 JoinCondition.PrimaryKey

*Applies to [RDL 2011/01](#)*

The **JoinCondition.PrimaryKey** element specifies an expression to evaluate in the [DefaultRelationship.RelatedDataSet](#). The **JoinCondition.PrimaryKey** element MUST be specified and MUST NOT be specified more than once. The value of this element MUST be an expression that evaluates to a **Variant**. The value of this element MUST NOT contain a reference to [Variable](#), **ReportItem**, **RunningValue**, **Previous**, or any aggregate functions. [<181>](#)

Following is the parent element of the **JoinCondition.PrimaryKey** element.

<b>Parent elements</b>
------------------------

<a href="#">JoinCondition</a>
-------------------------------

The following is the XML Schema definition of the **JoinCondition.PrimaryKey** element.

```
<xsd:element name="PrimaryKey" type="xsd:string" />
```

### 2.338.3 JoinCondition.SortDirection

**Applies to** [RDL 2011/01](#)

The **JoinCondition.SortDirection** element specifies the sort order of the [JoinCondition](#) element. The **JoinCondition.SortDirection** element is optional and MUST NOT be specified more than once. If the **JoinCondition** has a containing [Relationship](#), **JoinCondition.SortDirection** MUST NOT be specified unless [Relationship.NaturalJoin](#) is true. If the **JoinCondition** has a containing [DefaultRelationship](#), **JoinCondition.SortDirection** MUST NOT be specified unless **DefaultRelationship.NaturalJoin** is true. If the **JoinCondition.SortDirection** element is specified, its value MUST be one of the following:

**Ascending** (default): Both [DataSets](#) involved in the **data correlation** are sorted in ascending order by the **JoinCondition**.

**Descending**: Both **DataSets** involved in the data correlation are sorted in descending order by the **JoinCondition**.

If the **JoinCondition.SortDirection** element is not specified, its value is interpreted as "Ascending".

Following is the parent element of the **JoinCondition.SortDirection** element.

<b>Parent elements</b>
------------------------

<a href="#">JoinCondition</a>
-------------------------------

The following is the XML Schema definition of the **JoinCondition.SortDirection** element.

```
<xsd:element name="SortDirection" minOccurs="0" maxOccurs="1">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Ascending"/>
      <xsd:enumeration value="Descending"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
```

### 2.339 Relationships

**Applies to** [RDL 2011/01](#)

The **Relationships** element specifies a collection of [Relationship](#) elements.

The following are the parent and child elements of the **Relationships** element.

<b>Parent elements</b>
------------------------

<a href="#">TablixCell</a>
----------------------------

Parent elements
<a href="#">ChartDataPoint</a>
<a href="#">DataCell</a>

Child elements
<a href="#">Relationships.Relationship</a>

The following is the XML Schema definition of the **Relationships** element.

```
<xsd:complexType name="RelationshipsType">
  <xsd:sequence>
    <xsd:element name="Relationship" type="RelationshipType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.339.1 Relationships.Relationship

**Applies to** [RDL 2011/01](#)

The **Relationships.Relationship** element specifies a relationship to use for correlating data in the current scope with data in a containing scope. This element **MUST** be specified at least once and can be specified more than once. The **Relationships.Relationship** element is of type [Relationship](#).

Following is the parent element of the **Relationships.Relationship** element.

Parent elements
<a href="#">Relationships</a>

The following is the XML Schema definition of the **Relationships.Relationship** element.

```
<xsd:element name="Relationship" type="RelationshipType" />
```

### 2.340 Relationship

**Applies to** [RDL 2011/01](#)

The **Relationship** element specifies a **relationship** to use for correlating data in the current **scope** with data in a containing scope. [<182><183><184><185>](#)

The following are the parent and child elements of the **Relationship** element.

Parent elements
<a href="#">Group</a>
<a href="#">Relationships</a>
<a href="#">Chart</a>

Parent elements
<a href="#">MapDataRegion</a>
<a href="#">GaugePanel</a>
<a href="#">CustomData</a>
<a href="#">Tablix</a>

Child elements
<a href="#">Relationship.ParentScope</a>
<a href="#">Relationship.NaturalJoin</a>
<a href="#">Relationship.JoinConditions</a>

The following is the XML Schema definition of the **Relationship** element.

```
<xsd:complexType name="RelationshipType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ParentScope" type="xsd:string" minOccurs="0" />
    <xsd:element name="NaturalJoin" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="JoinConditions" type="JoinConditionsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

### 2.340.1 Relationship.ParentScope

**Applies to** [RDL 2011/01](#)

The **Relationship.ParentScope** element specifies the name of the target-related scope (**DataRegion** or [Group](#)) for this relationship.

The **Relationship.ParentScope** element is optional and MUST NOT be specified more than once. If this element is present, its value MUST be a [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1).

Within an **intersection scope**, **Relationship.ParentScope** is required and MUST reference the parent scope along one hierarchy of the containing **DataRegion**. Multiple **Relationship.ParentScope** elements MUST NOT specify the same value within the containing [Relationships](#) element.

**Relationship.ParentScope** is ignored for relationships that are outside of an intersection scope.

Parent elements
<a href="#">Relationship</a>

The following is the XML Schema definition of the **Relationship.ParentScope** element.

```
<xsd:element name="ParentScope" type="xsd:string" minOccurs="0" />
```



## 2.340.2 Relationship.NaturalJoin

*Applies to [RDL 2011/01](#)*

The **Relationship.NaturalJoin** element specifies that the dataset associated with current scope in the relationship is in the same order as the dataset associated with the target scope in the relationship with respect to the **join** condition. The **Relationship.NaturalJoin** element is optional and MUST NOT be specified more than once. If this element is present, its value MUST be a [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2). If this element is not present, its value is interpreted as false. <186>

Following is the parent element of the **Relationship.NaturalJoin** element.

Parent elements
<a href="#">Relationship</a>

The following is the XML Schema definition of the **Relationship.NaturalJoin** element.

```
<xsd:element name="NaturalJoin" type="xsd:boolean" minOccurs="0" />
```

## 2.340.3 Relationship.JoinConditions

*Applies to [RDL 2011/01](#)*

The **Relationship.JoinConditions** element specifies the **join** conditions that specify which instances of this scope (**DataRegion** or [Group](#)) correspond to which instance of the [Relationship.ParentScope](#). The **Relationship.JoinConditions** element is optional and MUST NOT be specified more than once. This element is of type [JoinConditions](#). <187>

Parent elements
<a href="#">Relationship</a>

The following is the XML Schema definition of the **Relationship.JoinConditions** element.

```
<xsd:element name="JoinConditions" type="JoinConditionsType" minOccurs="0" />
```

## 2.341 Expressions

### 2.341.1 Expression Syntax

In addition to supporting constants as element values, RDL supports an expression language. All expressions in RDL MUST begin with the character "=" and MUST be specified in a Visual Basic-compatible syntax [MSFT-VBNET].

Expression values that do not begin with "=" are treated as constants of the type expected by the property if that type is [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2), [String](#) ([XMLSCHEMA2/2] section 3.2.1), or [Integer](#) ([XMLSCHEMA2/2] section 3.3.17).

For example, the [Visibility.Hidden](#) property expects a **Boolean**. Therefore, the strings true and false are treated as **Boolean** constants. For properties that take a **VARIANT**, all values that do not start with "=" are treated as string constants.

## 2.341.2 Custom Code References

A report can include references to custom code methods and custom assemblies. Custom assemblies that are referenced in a report MUST be specified at the report level via the [CodeModules](#) element. Static methods in custom assemblies MAY be accessed globally from expressions within the report, and, if referenced from expressions, MUST use the following syntax.

```
ClassName.MethodName (...)
```

The following is an example of this syntax.

```
MyCurrencyConverterClass.Convert (...)
```

A report can contain instance-based methods. Such methods MUST be instantiated through the [Classes](#) element. The methods MAY be accessed via a globally available **Code** member, and they MUST use the following syntax.

```
Code.InstanceName.MethodName (...)
```

The following is an example of this syntax.

```
Code.CurrencyConverter.Convert (...)
```

## 2.341.3 Built-in Code Namespace References

In addition to explicit user-defined references as specified through custom code references, expressions MAY [<188>](#) reference function libraries if they are made available by the implementation.

## 2.341.4 Data Types

Every expression that is used in an expression element or as an argument to an RDL function MUST return one of the following RDL data types: [String](#) ([XMLSCHEMA2/2] section 3.2.1), [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2), [Float](#) ([XMLSCHEMA2] section 3.2.4), [DateTime](#) ([XMLSCHEMA2] section 3.2.7), Binary, or an array of these types (a [VariantArray](#)). [<189>](#)

## 2.341.5 Expression Error Handling

Errors that occur during expression evaluation fall into one of two categories:

- Critical errors
  - Critical errors result in the report (or the currently requested page) failing to render, returning an error message instead.
  - Errors in the evaluation of the following properties are considered to be critical errors: [Filter.FilterExpression](#), [Filter.FilterValues](#), [Group.GroupExpressions](#), [Variable.Value](#), [Visibility.Hidden](#), and [SortExpression.Value](#).
- Non-critical errors
  - A non-critical error does not cause the report to fail to render but instead registers a warning and falls back to a default value for the particular RDL property.

## 2.341.6 Global Collections

The following global object collections are accessible from expressions, with restrictions as specified in the following subsections for each collection.

Collection	Specification	Item type
<a href="#">Fields</a>	Fields in the current dataset.	<a href="#">Field</a>
<a href="#">Parameters</a>	Report parameters.	<a href="#">Parameter</a>
<a href="#">ReportItems</a>	All text boxes in the report.	<b>ReportItem</b>
<a href="#">Globals</a>	Global variables.	<b>Variant</b>
<a href="#">User</a>	User-specific data.	<b>Variant</b>

**Applies to** [RDL 2012/01](#)

Collection	Specification	Item type
<a href="#">Scopes</a>	Scopes within the report.	<a href="#">Scope</a>

Data sources in **global collections**, if referenced from expressions, MUST use standard Visual Basic collection syntax [MSFT-VBNET], specified as one of the following:

*Collection!ObjectName*

*Collection.Item("ObjectName")*

*Collection("ObjectName")*

such as in the following example:

```
User!Language
```

Optionally, items in the **Globals** and **User** collections can also be accessed by using property syntax:

```
Collection.ObjectName
```

such as in:

```
Globals.PageNumber
```

Circular references involving items in global collections are errors.

### 2.341.6.1 Fields

The **Fields** collection contains a set of [Field](#) objects. A **Field** object can specify one or more properties. If one of these properties is referenced in an expression, it MUST be among the set of predefined properties that can be accessed by using either the property syntax:

```
Fields!FieldName.PropertyName
```

or the collection syntax:

```
Fields!FieldName!PropertyName
```

```
Fields!FieldName("PropertyName")
```

```
Fields!FieldName.Properties("PropertyName")
```

such as:

```
Fields!Region.BackgroundColor
```

## 2.341.6.2 Predefined Field Properties

When a report is executed, dataset queries can return a different set of fields than were originally defined in the report. The **IsMissing** property specifies whether the field was found in the resulting dataset. The **Value** property of missing fields is NULL.

In addition, data providers that support field properties can provide additional properties, which can be accessed only by using collection syntax. If the data provider does not support the requested property or if the field is not found when the query is executed, the default value returned is NULL for **String Object** properties, 0 for [Integer](#) ([XMLSCHEMA2/2] section 3.3.17) properties, and false for [Boolean](#) ([XMLSCHEMA2/2] section 3.2.2) properties. The value of all predefined field properties is the data type specified in the following table. The value of all other properties is a **Variant**.

The following are all predefined properties of the [Field](#) object.

Property name	Type	Expected Values
<b>Value</b>	<b>Variant</b>	
<b>IsMissing</b>	<b>Boolean</b>	
<b>UniqueName</b>	<a href="#">String</a> ([XMLSCHEMA2/2] section 3.2.1)	
<b>BackgroundColor</b>	<b>String</b>	See <a href="#">Style.BackgroundColor</a>
<b>Color</b>	<b>String</b>	See <a href="#">Style.Color</a>
<b>FontFamily</b>	<b>String</b>	See <a href="#">Style.FontFamily</a>
<b>FontSize</b>	<b>String</b>	See <a href="#">Style.FontSize</a>
<b>FontWeight</b>	<b>String</b>	See <a href="#">Style.FontWeight</a>
<b>FontStyle</b>	<b>String</b>	See <a href="#">Style.FontStyle</a>
<b>TextDecoration</b>	<b>String</b>	See <a href="#">Style.TextDecoration</a>
<b>FormattedValue</b>	<b>String</b>	
<b>Key</b>	<b>Variant</b>	
<b>LevelNumber</b>	<b>Integer</b>	
<b>ParentUniqueName</b>	<b>String</b>	

### 2.341.6.3 Fields Collection in Reports with Multiple Datasets

If a [Report](#) contains multiple datasets, there are multiple virtual [Fields](#) collections in the report. The context of the expression usage in the report MUST unambiguously specify which of these is accessed when referencing the **Fields** collection as one of the following three possible options:

- Inside of an aggregate, if the *Scope* argument refers to a dataset, **Fields** refers to the fields in that dataset.
- Within a data region, the **Fields** collection refers to the fields in the dataset for that region.
- Outside of a data region, direct references to fields (outside an aggregate) are undefined.

### 2.341.6.4 Parameters

The **Parameters** collection contains a set of [Parameter](#) objects (one **Parameter** object per [ReportParameter](#) specified in the report). The **Parameter** object specifies a set of predefined properties that, if referenced from an expression, MUST be accessed by using either the property syntax or the collection syntax (see [Fields](#)).

The only properties specified for **Parameter** are **Value**, **Label**, **Count**, and **IsMultiValue** as specified in the following table:

Property name	Type	Specification
<b>Value</b>	<b>Variant</b> <b>VariantArray</b>	The <b>Value</b> property is determined by the parameter value input provided to the current report execution session. The data type of <b>Value</b> is a <b>Variant</b> for single-value parameters and <b>VariantArray</b> for multivalued parameters.
<b>Label</b>	<a href="#">String</a> ([XMLSCHEMA2/2] section 3.2.1) <b>StringArray</b>	The <b>Label</b> property is determined by the value of the <b>Label</b> or <b>LabelField</b> element of the corresponding parameter value in the <b>ValidValues</b> list. The data type of <b>Label</b> is a <b>String</b> for single-value parameters and <b>StringArray</b> for multivalued parameters.
<b>Count</b>	<a href="#">Integer</a> ([XMLSCHEMA2/2] section 3.3.17)	The <b>Count</b> property specifies the number of values and labels. The value of the <b>Count</b> property is 1 if it is not a multivalued parameter; it is 0 if the parameter does not have a value that is valid (or values have not yet been supplied).
<b>IsMultiValue</b>	<a href="#">Boolean</a> ([XMLSCHEMA2/2] section 3.2.2)	The <b>IsMultiValue</b> property specifies whether the parameter is multivalued. The value of this property is <b>Boolean</b> .

### 2.341.6.5 ReportItems

Only text boxes appear as **ReportItem** objects in the **ReportItems** collection. <190>

The **ReportItem** object has a collection of predefined properties that, if referenced from expressions, MUST be referenced by using either the property syntax or the collection syntax (see the [Fields](#) object).

The only property defined for **ReportItem** is **Value**. The data type of **Value** is **Variant**.

The value of the current **ReportItem** can be referenced in property expressions by using **Me.Value** or simply **Value**. Aggregate functions MUST NOT include a reference to **Me.Value** or **Value**.

Expressions SHOULD NOT reference report items that are not defined at or above the current **grouping scope** (such as peer or descendent grouping scopes). If such a reference is used, the result is undefined.

The scope for expressions in page headers and page footers is considered to be items on the current page.

### 2.341.6.6 Globals

Members of the **Globals** collection are of type **VARIANT**, but individual members have the following known return types.

Property name	Type	Specification
<b>PageNumber</b>	<a href="#">Integer</a> ( <a href="#">[XMLSCHEMA2/2]</a> section 3.3.17)	Current page number. The member MUST NOT be referenced outside of the page header or page footer of the report.
<b>TotalPages</b>	<b>Integer</b>	Total number of pages in the report. The member MUST NOT be referenced outside of the page header or page footer of the report.
<b>ExecutionTime</b>	<a href="#">DateTime</a> ( <a href="#">[XMLSCHEMA2]</a> section 3.2.7)	The date and time at which the report began executing.
<b>ReportServerUrl</b>	<a href="#">String</a> ( <a href="#">[XMLSCHEMA2/2]</a> section 3.2.1)	URL to the report server, such as <a href="http://reportserver/reports">http://reportserver/reports</a> .
<b>ReportFolder</b>	<b>String</b>	Full path on the report server to the folder containing the report. For example, for the report <a href="http://reportserver/reports/salesreports/budgeting/currentbudget">http://reportserver/reports/salesreports/budgeting/currentbudget</a> , the <b>ReportFolder</b> is <a href="#">/salesreports/budgeting</a> .
<b>ReportName</b>	<b>String</b>	Name of the report in the report catalog. For example, for the report <a href="http://reportserver/reports/salesreports/budgeting/currentbudget">http://reportserver/reports/salesreports/budgeting/currentbudget</a> , the <b>ReportName</b> is "currentbudget".
<b>RenderFormat.Name</b>	<b>String</b>	Name of the renderer. <a href="#">&lt;191&gt;</a>
<b>RenderFormat.IsInteractive</b>	<a href="#">Boolean</a> ( <a href="#">[XMLSCHEMA2/2]</a> section 3.2.2)	Specifies whether the current user request is for an interactive rendering format.
<b>RenderFormat.DeviceInfo</b>	<a href="#">ReadOnlyNameValueCollection</a>	A collection that holds the keys and values of the <b>deviceinfo</b> parameters of the current rendering request. <a href="#">&lt;192&gt;</a>

### 2.341.6.7 ReadOnlyNameValueCollection

A **ReadOnlyNameValueCollection** collection is identical to the **NameValueCollection** [MSFT-VBNET] class of the Microsoft .NET Framework, with the following exceptions:

- The collection is read-only. Only the following class get-properties and methods exist:

- **Properties:** **AllKeys**, **this[int index]**, **this[string name]**, **Count**, **Keys**
- **Methods:** **CopyTo**, **Get**, **GetKey**, **GetValues**, **HasKeys**, **GetEnumerator**
- An additional class get-property called **AllValues** exists. This property gets all values in the entire read-only name value collection and returns a **StringArray** of those values.

### 2.341.6.8 User

Members of the **User** collection are of type **Variant**, but individual members have the following known return types.

Property name	Type	Specification
<b>UserID</b>	<a href="#">String</a> ( <a href="#">XMLSCHEMA2/2</a> section 3.2.1)	ID of the user executing the report.
<b>Language</b>	<b>String</b>	Language ID of the client executing the report.

### 2.341.6.9 DataSources

The **DataSources** collection contains a collection of [DataSource](#) objects. Only data sources that are used in the [Body](#) of the report are included in the **DataSources** collection. Data sources that are used only in parameter valid values and default values properties are not included.

The **DataSource** object specifies the following properties.

Property name	Type	Specification
<b>Type</b>	<a href="#">String</a> ( <a href="#">XMLSCHEMA2/2</a> section 3.2.1)	Type of <b>data provider</b> for the data source.
<b>DataSourceReference</b>	<b>String</b>	Path to the data source ( <b>Nothing</b> for embedded data sources).

### 2.341.6.10 DataSets

The **DataSets** collection contains a collection of [DataSet](#) objects. Only datasets used in the [Body](#) of the report are included in the **DataSets** collection. Datasets that are used only in parameter valid values and default values properties are not included.

The **DataSet** object specifies the following properties.

Property name	Type	Specification
<b>CommandText</b>	<a href="#">String</a> ( <a href="#">XMLSCHEMA2/2</a> section 3.2.1)	The <b>CommandText</b> of the dataset.
<b>RewrittenCommandText</b>	<b>String</b>	The <b>CommandText</b> of the dataset after optionally rewritten by the report execution engine.

### 2.341.6.11 Variables

The **Variables** collection contains a collection of [Variable](#) objects (one object each per variable defined on report element and group elements).

The **Variable** object has a collection of predefined properties that, if referenced from expressions, MUST be accessed via either the property syntax or the collection syntax (see [Fields](#)).

The **Variable** object specifies the properties that are described in the following table.

Property name	Type	Specification
Value	Serializable	Gets the value of the report or group variable.Sets the value of the report variable, if the report variable is writable.
Writable	Boolean	Indicates whether the variable is writable through expressions.

The **Variable** object specifies the method that is described in the following table.

Method name and argument type	Return type	Specification
<b>SetValue(Serializable)</b>	Boolean	Sets the value of a report variable. Returns true if setting the value was successful. Returns false for group variables and for non-writable report variables.

Expressions SHOULD NOT reference variables that are not defined at or above the current **grouping scope** (such as peer or descendent grouping scopes).

### 2.341.6.12 Restrictions on Use of Global Collections

The [Fields](#), [Parameters](#), [ReportItems](#), and [Globals](#) collections have restrictions on the contexts in which they can be used in expressions. The following table specifies where these **global collections** can ("Yes," or with restrictions specified below) or cannot ("No") be used.

Usage Context	Global collections						
	Fields	ReportItems	Parameters	PageNumber TotalPages	DataSource DataSet	Variables	Scopes
PageHeader or PageFooter	Yes	Yes*	Yes	Yes	Yes	Yes	No
Body	Yes*	Yes***	Yes	No	Yes	Yes	Yes***** *
ReportParameter	No	No	Yes****	No	No	No	No
Field	Yes	No	Yes	No	No< <a href="#">193</a> >	No	No
Query Parameter	No	No	Yes	No	No	No	No
Group Expression	Yes	No	Yes	No	Yes	No	No



<b>Usage</b>	<b>Global collections</b>						
Sort Expression	Yes	No	Yes	No	Yes	Yes*****	No
Filter Expression	Yes	No	Yes	No	Yes	Yes***** *	No
Code	No	No	Yes	No	No	No	No
Report.Language	No	No	Yes	No	No	No	No
Variables	Yes	No	Yes	No	Yes	Yes*****	No
Aggregates	Yes	Yes***** *	Yes	Yes***** *	Yes	No	No
Lookup LookupSet	Yes	Yes	Yes	Yes	Yes	No	No***** **

\* MUST NOT contain more than one report item reference.

\*\* If an expression in a non-detail section refers to a field that is not in the group expression for the group (or any ancestor **grouping scope**), which specific value is used is not defined. The expression SHOULD use the First() and Last() aggregate functions.

\*\*\* MUST NOT contain report item references other than those in the current or ancestor **scope**.

\*\*\*\* MUST NOT contain parameter references to subsequent parameters.

\*\*\*\*\* Variable references in sort expressions can appear in [TablixMember](#), [DataMember](#), and [ChartMember](#) elements. Variable references MUST NOT appear in **DataRegion.SortExpressions**.

\*\*\*\*\* MUST NOT contain variable references other than those in the current or ancestor scope.

\*\*\*\*\* MUST NOT be used outside a page header or page footer.

\*\*\*\*\* MUST NOT be used outside report item aggregates in a page header or page footer.

\*\*\*\*\* Variable references in filter expressions can appear in **TablixMember**, **DataMember**, and **ChartMember** elements. Variable references MUST NOT appear in **DataRegion.Filters** or **DataSet.Filters**.

\*\*\*\*\* MUST NOT be used for **ReportItems** outside **DataRegion**.

\*\*\*\*\* can only be used in the **Source** expression of a **Lookup** function.

Note that, because references to items in global collections can be dynamic, such as in "=ReportItems(Parameters!Param1.Value)", all error checking occurs both during **report** validation (to catch static disallowed references) and during report execution (to catch dynamic disallowed references).

Cyclic expressions (such as TextBox1=TextBox2+1; TextBox2=TextBox1+1) are treated as non-critical errors.

### 2.341.6.13 Scopes

**Applies to** [RDL 2012/01](#)

The **Scopes** collection contains a collection of **Scope** objects. The **Scopes** collection is indexed by **scope** name and is bound to the [DataSet](#) in the report that has a [DefaultRelationship](#) to the current scope or to one of its ancestor scopes.

The **Scope** object specifies the following property.

Property name	Type	Specification
<a href="#">Fields</a>	Fields	The <b>Fields</b> from the <b>DataSet</b> to which the scope is bound.

The **Scope** object has a **Fields** property, which contains the **Fields** from the **DataSet** to which the scope is bound.

The **Scopes** collection can only be used in expressions that have the following form.

```
Scopes!ScopeName.Fields!FieldName.Value
```

### 2.341.7 Aggregate Functions

RDL specifies the following standard **aggregate functions**:

- **Sum**
- **Avg**
- **Max**
- **Min**
- **Count**
- **CountDistinct**
- **CountRows**
- **StDev**
- **StDevP**
- **Var**
- **VarP**

In addition, RDL specifies the following advanced aggregate functions:

- **First**
- **Last**
- **Previous**
- **RunningValue**
- **RowNumber**
- **Aggregate**

Common parameters used by aggregate functions are specified in the following [Scope](#) and [Recursive](#) sections. A detailed specification for each aggregate function follows in subsequent sections.

### 2.341.7.1 Scope

There are three types of explicit **scopes** for expressions. These types are as follows:

- [DataSet](#) scope.
- **DataRegion** scope.
- [Grouping](#) scope.

Explicit scopes (**DataSets**, **DataRegions**, and **Groups**) MUST have unique names.

In addition, there are three types of implicit scopes that do not have names:

- Top-level [Report](#) scope.
- [Page](#) scope.
- Cell scope (the intersection of the innermost row scope and innermost column scope).

Scopes represent a hierarchy. At any level in that hierarchy, there can be only one ancestor (except for the top-level report scope and the page scope) but an unlimited number of descendants as well as peer scopes. The "innermost scope" is specified as the position within that scope hierarchy relative to the current RDL expression.

The *Scope* parameter of an aggregate function MUST be a constant.

Three different report areas are distinguished for the *Scope* parameter.

For expressions inside **data regions**:

- Within a data region, the *Scope* argument can be omitted for all aggregates with the exception of [RunningValue](#) and [RowNumber](#).
- If omitted, the scope is determined as the innermost scope containing the report item in which the aggregate is used.

For expressions outside of data regions (in the report [Body](#)):

- When used outside of a data region, the scope, if specified, MUST refer to a dataset name.
- If more than one dataset exists in the report, the *Scope* argument MUST be specified.
- If exactly one dataset exists in the report, the *Scope* argument can be omitted.

For expressions in page headers and page footers:

- If the *Scope* argument is omitted in page headers and page footers, the scope is the data on the current page. In this case, report items can be used in an aggregate expression, but references to fields MUST NOT appear in the aggregate expression.
- If a scope is specified, fields can be used, but report items MUST NOT be used.

### 2.341.7.2 Recursive

Specifying an aggregate as recursive indicates that the aggregate applies to all data in the current instance of the given scope and all descendant instances of the current instance.

Recursive MAY be specified on standard aggregate functions: **Sum**, **Avg**, **Max**, **Min**, **Count**, **CountDistinct**, **CountRows**, **StDev**, **StDevP**, **Var**, or **VarP**.

Recursive SHOULD NOT be specified if the scope has no **Parent** property value specified on the **group expression**; if it is specified, it is ignored.

Recursive MUST NOT be specified if the aggregate function is nested inside another aggregate function.

Recursive MUST NOT be specified if the aggregate function contains other aggregate functions.

The following table is an example of data aggregations, calculated without taking into account the recursive hierarchy ("Sales" column), and calculated with the recursive grouping based on ManagerID ("AllSales" column).

EmployeeID	ManagerID	Sales (Non Recursive)	AllSales (Recursive with Parent = ManagerID)
1	NULL	10	70
1a	1	10	30
1a1	1a	10	10
1a2	1a	10	10
1b	1	10	30
1b1	1b	10	10
1b2	1b	10	10

### 2.341.7.3 Recursive Depth

In recursive hierarchies, the **Level** function can be used to determine the current depth of the recursive hierarchy.

### 2.341.7.4 Function: Level

The following are the arguments, types, and specification of the **Level** function.

Function	Arguments	Type	Specification
<b>Level</b>	<i>Return</i>	<b>Integer</b>	A zero-based <b>Integer</b> specifying the current depth level of a recursive hierarchy. If the specified scope is a dataset, data region, or group without a parent, or if the scope does not exist, <b>Level</b> returns 0.
	<i>Scope</i>	<b>String</b>	Optional. Defaults to the current scope.

### 2.341.7.5 Aggregate Function: Sum

The following are the arguments, types, and specification of the **Sum** aggregate function.

Function	Arguments	Type	Specification
<b>Sum</b>	<i>Return</i>	<a href="#">Float</a>	Returns the sum of all values of the expression within the scope. <i>Return</i> type is decimal for decimal* expressions and double for all other

Function	Arguments	Type	Specification
			expressions.
	<i>Expression</i>	<b>Numeric</b>	The expression to aggregate. MUST NOT contain the <b>First</b> , <b>Last</b> , <b>Previous</b> , or <b>RunningValue</b> functions. MUST NOT reference other aggregate functions unless their <i>Scope</i> is the same as, or a child scope of, this function's <i>Scope</i> . For all distinct scopes directly** referenced in the expression, there MUST be one scope that has all other scopes as its ancestor.  Aggregate functions in the page header or page footer with an omitted <i>Scope</i> parameter MUST NOT contain any aggregate functions.  Aggregate functions with the name of a dataset as its <i>Scope</i> parameter MUST NOT contain any aggregate functions.
	<i>Scope</i>	<a href="#">String</a>	Name of a dataset or the name of a group or data region that contains (directly or indirectly) the report item that the aggregate function is used in. Specifies that the aggregate applies to the entire dataset, all of the data in the current group, or all of the data in the current data region. MUST be a constant, not an expression. See also <a href="#">Scope</a> .
	<i>Recursive</i>	<b>Enum</b>	Recursive   Simple (Default). Specifies whether the aggregate is calculated recursively. Optional. See also <a href="#">Recursive</a> .

\* Decimal and double refer to CLR data types in this context.

\*\* Directly referenced aggregate functions are contained within the *Expression* argument but are not contained within a nested aggregate or the *Destination* or *Result* arguments to a **Lookup** function.

### 2.341.7.6 Aggregate Function: Avg

The following are the arguments, types, and specification of the **Avg** aggregate function.

Function	Arguments	Type	Specification
<b>Avg</b>	<i>Return</i>	<a href="#">Float</a>	Returns the average of all non-null values of the expression within the specified scope. For the <i>Return</i> type, see the <a href="#">Sum</a> function.
	<Expression>	<b>Numeric</b>	See the <b>Sum</b> function.
	<i>Scope</i>	<a href="#">String</a>	See the <b>Sum</b> function.
	<i>Recursive</i>	<b>Enum</b>	See the <b>Sum</b> function.

### 2.341.7.7 Aggregate Function: Max

The following are the arguments, types, and specification of the **Max** aggregate function.

Function	Arguments	Type	Specification
<b>Max</b>	<i>Return</i>	<b>Variant</b>	Returns the maximum of all non-null values of the expression within the specified scope. <i>Return</i> type is the same as the expression type.
	<i>Expression</i>	<b>Variant</b>	See the <a href="#">Sum</a> function.

Function	Arguments	Type	Specification
	<i>Scope</i>	<a href="#">String</a>	See the <b>Sum</b> function.
	<i>Recursive</i>	<b>Enum</b>	See the <b>Sum</b> function.

### 2.341.7.8 Aggregate Function: Min

The following are the arguments, types, and specification of the **Min** aggregate function.

Function	Arguments	Type	Specification
<b>Min</b>	<i>Return</i>	<b>Variant</b>	Returns the minimum of all non-null values of the expression within the specified scope. <i>Return</i> type is the same as the expression type.
	<i>Expression</i>	<b>Variant</b>	See the <a href="#">Sum</a> function.
	<i>Scope</i>	<a href="#">String</a>	See the <b>Sum</b> function.
	<i>Recursive</i>	<b>Enum</b>	See the <b>Sum</b> function.

### 2.341.7.9 Aggregate Function: Count

The following are the arguments, types, and specification of the **Count** aggregate function.

Function	Arguments	Type	Specification
<b>Count</b>	<i>Return</i>	<a href="#">Integer</a>	Returns the count of all non-null values of the expression within the specified scope.
	<i>Expression</i>	<b>Variant</b> or <b>Binary</b>	See the <a href="#">Sum</a> function.
	<i>Scope</i>	<a href="#">String</a>	See the <b>Sum</b> function.
	<i>Recursive</i>	<b>Enum</b>	See the <b>Sum</b> function.

### 2.341.7.10 Aggregate Function: CountDistinct

The following are the arguments, types, and specification of the **CountDistinct** aggregate function.

Function	Arguments	Type	Specification
<b>CountDistinct</b>	<i>Return</i>	<a href="#">Integer</a>	Returns the count of all distinct non-null values of the expression within the specified scope.
	<i>Expression</i>	<b>Variant</b>	See the <a href="#">Sum</a> function.
	<i>Scope</i>	<a href="#">String</a>	See the <b>Sum</b> function.

Function	Arguments	Type	Specification
	<i>Recursive</i>	<b>Enum</b>	See the <b>Sum</b> function.

### 2.341.7.11 Aggregate Function: CountRows

The following are the arguments, types, and specification of the **CountRows** aggregate function.

Function	Arguments	Type	Specification
<b>CountRows</b>	<i>Return</i>	<a href="#">Integer</a>	Returns the count of all rows within the specified scope.
	<i>Scope</i>	<a href="#">String</a>	See the <a href="#">Sum</a> function.
	<i>Recursive</i>	<b>Enum</b>	See the <b>Sum</b> function.

### 2.341.7.12 Aggregate Function: StDev

The following are the arguments, types, and specification of the **StDev** aggregate function.

Function	Arguments	Type	Specification
<b>StDev</b>	<i>Return</i>	<a href="#">Float</a>	Returns the standard deviation of all non-null values of the expression within the specified scope.
	<i>Expression</i>	<b>Numeric</b>	See the <a href="#">Sum</a> function.
	<i>Scope</i>	<a href="#">String</a>	See the <b>Sum</b> function.
	<i>Recursive</i>	<b>Enum</b>	See Sum.

### 2.341.7.13 Aggregate Function: StDevP

The following are the arguments, types, and specification of the **StDevP** aggregate function.

Function	Arguments	Type	Specification
<b>StDevP</b>	<i>Return</i>	<a href="#">Float</a>	Returns the population standard deviation of all non-null values of the expression within the specified scope. <i>Return</i> type is the same as the expression type.
	<i>Expression</i>	<b>Numeric</b>	See the <a href="#">Sum</a> function.
	<i>Scope</i>	<a href="#">String</a>	See the <b>Sum</b> function.
	<i>Recursive</i>	<b>Enum</b>	See the <b>Sum</b> function.

### 2.341.7.14 Aggregate Function: Var

The following are the arguments, types, and specification of the **Var** aggregate function.

Function	Arguments	Type	Specification
<b>Var</b>	<i>Return</i>	<a href="#">Float</a>	Returns the variance of all non-null values of the expression within the specified scope. See the <a href="#">Sum</a> function regarding the <i>return</i> type.
	<i>Expression</i>	<b>Numeric</b>	See the <b>Sum</b> function.
	<i>Scope</i>	<a href="#">String</a>	See the <b>Sum</b> function.
	<i>Recursive</i>	<b>Enum</b>	See the <b>Sum</b> function.

### 2.341.7.15 Aggregate Function: VarP

The following are the arguments, types, and specification of the **VarP** aggregate function.

Function	Arguments	Type	Specification
<b>VarP</b>	<i>Return</i>	<a href="#">Float</a>	Returns the population variance of all non-null values of the expression within the specified scope. See the <a href="#">Sum</a> function regarding the <i>return</i> type.
	<i>Expression</i>	<b>Numeric</b>	See the <b>Sum</b> function.
	<i>Scope</i>	<a href="#">String</a>	See the <b>Sum</b> function.
	<i>Recursive</i>	<b>Enum</b>	See the <b>Sum</b> function.

### 2.341.7.16 Aggregate Function: Union

The following are the arguments, types, and specification of the **Union** aggregate function.

Function	Arguments	Type	Specification
<b>Union</b>	<i>Return</i>	SqlGeography or SqlGeometry	Returns the union of all values of the expression within the scope. <i>Return</i> type is SqlGeography for SqlGeography expressions and SqlGeometry for SqlGeometry expressions.
	<i>Expression</i>	SqlGeography or SqlGeometry	See the <a href="#">Sum</a> function. All the expressions MUST be either of type SqlGeography or SqlGeometry.
	<i>Scope</i>	<a href="#">String</a>	See the <b>Sum</b> function.

### 2.341.7.17 Aggregate Function: First

The following are the arguments, types, and specification of the **First** aggregate function.



Function	Arguments	Type	Specification
<b>First</b>	<i>Return</i>	<b>Variant or Binary</b>	Returns the first value of the expression within the scope (after all sorting up through the <i>Scope</i> has been applied). <i>Return</i> type is the same as the expression type.
	<i>Expression</i>	<b>Variant or Binary</b>	See the <a href="#">Sum</a> function.
	<i>Scope</i>	<a href="#">String</a>	See the <b>Sum</b> function.

### 2.341.7.18 Aggregate Function: Last

The following are the arguments, types, and specification of the **Last** aggregate function.

Function	Arguments	Type	Specification
<b>Last</b>	<i>Return</i>	<b>Variant or Binary</b>	Returns the last value of the expression within the scope (after all sorting up through the <i>Scope</i> has been applied). <i>Return</i> type is the same as the expression type.
	<i>Expression</i>	<b>Variant or Binary</b>	See the <a href="#">Sum</a> function.
	<i>Scope</i>	<a href="#">String</a>	See the <b>Sum</b> function.

### 2.341.7.19 Aggregate Function: Previous

The following are the arguments, types, and specification of the **Previous** aggregate function.

Function	Arguments	Type	Specification
<b>Previous</b>	<i>Return</i>	<b>Variant or Binary</b>	Returns the value of the expression for the previous instance of <i>PreviousScope</i> , or (if the expression is an aggregate), returns the value of the aggregate expression as applied to the previous instance of the <i>PreviousScope</i> corresponding to the current instance of the <i>Scope</i> of the aggregate function. Returns NULL if there is no corresponding previous instance.
	<i>Expression</i>	<b>Variant or Binary</b>	The expression for which to retrieve the previous value. If the expression contains an aggregate, <b>Previous</b> aggregates the data within the previous instance of the <i>PreviousScope</i> that corresponds to the current instance of the aggregate's <i>Scope</i> . An aggregate <i>Scope</i> equal to the <i>PreviousScope</i> indicates that all of the data in the <i>PreviousScope</i> is aggregated. The scope of the aggregate MUST be contained by (or equal to) <i>PreviousScope</i> . The aggregate function MUST NOT be <b>Aggregate</b> or <b>Previous</b> . The aggregate MUST NOT be recursive. The functions <a href="#">Level</a> and <a href="#">InScope</a> MUST NOT be used in the expression.
	<i>PreviousScope</i>	<a href="#">String</a>	If specified, MUST be the name of a group or data region that contains (directly or indirectly) the report item that the aggregate function is used in. <b>Previous</b> retrieves the data in the previous instance of the group or data region. A <i>PreviousScope</i> of <b>Nothing</b> specifies that <b>Previous</b> retrieves the value of the expression for the previous detail

Function	Arguments	Type	Specification
			row of data. <i>PreviousScope</i> MUST be a constant, not an expression. Optional. Default: The current scope ( <b>Nothing</b> if in a <b>detail scope</b> ).

### 2.341.7.20 Aggregate Function: RunningValue

The following are the arguments, types, and specification of the **RunningValue** aggregate function.

Function	Argument	Type	Specification
<b>RunningValue</b>	<i>Return</i>	See the <i>Function</i> argument.*	A running aggregate of the expression, using the specified aggregate function.
	<i>Expression</i>	See <i>Function</i>	The expression to aggregate. MUST NOT contain the <b>First</b> , <b>Last</b> , <b>Previous</b> , or <b>RunningValue</b> functions. Can reference other aggregate functions if their <i>Scope</i> is the same as, or a child scope of, this function's <i>Scope</i> . For all distinct scopes directly** referenced in the expression, there MUST be one scope that has all other scopes as its ancestor.
	<i>Function</i>	<b>Enum</b>	Name of an aggregate function for which to calculate a running value (MUST NOT be <b>CountRows</b> , <b>RunningValue</b> , <b>RowNumber</b> , or <b>Aggregate</b> ). <i>Expression</i> type and <i>Return</i> type are determined by the aggregate function used.
	<i>Scope</i>	<a href="#">String</a>	If specified, MUST be "Nothing", or the name of a group or data region that contains (directly or indirectly) the report item that the aggregate function is used in. The scope specifies that the running value is either reset whenever the group expression changes or reset with each new instance of the data region. A value of "Nothing" specifies that the running value never resets. The <i>Scope</i> argument MUST be a constant, not an expression.

\* The *Return* data type is determined by the function that is used in the *FunctionEnum* argument of the **RunningValue** call. For example, if **Sum** is used as *FunctionEnum* value for **RunningValue**, the return type follows the rules for the **Sum** function.

\*\* Directly referenced aggregate functions are contained within the *Expression* argument but are not contained within a nested aggregate or the *Destination* or *Result* arguments to a **Lookup** function.

### 2.341.7.21 Aggregate Function: RowNumber

The following are the arguments, types, and specification of the **RowNumber** aggregate function.

Function	Arguments	Type	Specification
<b>RowNumber</b>	<i>Return</i>	<a href="#">Integer</a>	The row number of the current row or group instance.
	<i>Scope</i>	<a href="#">String</a>	See <a href="#">RunningValue</a> .

## 2.341.7.22 Aggregate Function: Aggregate

The following are the arguments, types, and specification of the **Aggregate** aggregate function.

Function	Arguments	Type	Specification
<b>Aggregate</b>	<i>Return</i>	Determined by data provider.	Calculates a custom (data provider-defined) aggregate for the expression at the given scope. If the data provider does not support this function or if the data is not available for the given expression or scope, NULL is returned.
	<i>Expression</i>	<b>VARIANT</b>	The expression to aggregate. MUST be a simple field reference, for example: =Aggregate(Fields!Sales.Value,Year)
	<i>Scope</i>	<a href="#">String</a>	See the <a href="#">Sum</a> function. All group expressions for the <i>Scope</i> specified (and all containing group scopes) MUST be simple field references or (non-expression) constants.

## 2.341.7.23 Restrictions on Aggregate Usage

The following table specifies in which context a particular aggregate function call can ("Yes", or with restrictions specified below) or cannot ("No") be used.

Usage Context	Aggregation / Special Function Call						
	Running Value	Row Number	First / Last	Previous	Other Aggregate Functions	Report-Item Aggs	Lookup / LookupSet / MultiLookup
PageHeader or PageFooter	No	No	Yes	No	Yes	Yes	Yes
Body	Yes	Yes	Yes	Yes	Yes	No	Yes
ReportParameter	No	No	No	No	No	No	No
Calculated Field	No	No	No	No	No	No	No
Query Parameter	No	No	No	No	No	No	No
Group Expression	No	Yes	No	No	No	No	Yes
Sort Expression	No	No	No	No	Yes*	No	Yes
Tablix Cell	Yes**	Yes***	Yes	Yes	Yes	No	Yes
Group Variables	No	No	No	No	Yes****	No	Yes
Report Variables	No	No	No	No	Yes	No	Yes
DataSet.Filters	No	No	No	No	No	No	No
DataRegion.Filters	No	No	No	No	No	No	Yes
Group.Filters	No	No	No	No	Yes*****	No	Yes

\* MUST NOT appear in **DataRegion.SortExpressions**.

\*\* Within a **TablixCell** in the scope of both a **dynamic row** and a **dynamic column** of the tablix, the *Scope* argument for **RunningValue** and **RowNumber** MUST refer to either a column grouping or a row grouping for the tablix or to a scope contained within this **TablixCell**. The scope of the running value/row number specifies the direction of the running value. **HideDuplicates** in a **TablixCell** has the same restrictions/behavior. **RunningValue**, **RowNumber**, and **HideDuplicates** in cells of a tablix can specify either column scopes or row scopes, but not both.

\*\*\* MUST NOT appear in **DataRegion.SortExpressions**.

\*\*\*\* Aggregates that contain other aggregate functions MUST NOT appear in **Group.Variables**.

\*\*\*\*\* Aggregates that contain other aggregate functions MUST NOT appear in **Group.Filters**.

### 2.341.7.24 Restrictions on Aggregate Nesting

The following table specifies in which context a particular aggregate function call can ("Yes", or with restrictions specified below) or cannot ("No") be nested inside another aggregate or function call.

Usage Context	Aggregation / Special Function Call							
	Running Value	Row Number	First / Last	Previous	Other Aggregate Functions	Report-Item Aggs	Lookup / LookupSet/ MultiLookup	Aggregate
RunningValue	No	No	No	No	Yes	No	Yes	No
First / Last	No	No	No	No	Yes	No	No	No
Previous	Yes	Yes	Yes	No	Yes	No	Yes	No
Other Aggregate Functions	No	No	No	No	Yes	No	Yes	No
Report Item Aggregates	No	No	No	No	No	No	No	No
Lookup / LookupSet / MultiLookup	Yes*	Yes*	Yes*	Yes*	Yes*	Yes*	No	No
Aggregate	No	No	No	No	No	No	No	No

\* Aggregate functions MUST NOT be referenced inside the *Destination* or *Result* expressions of a **Lookup**, **LookupSet**, or **MultiLookup** function. Aggregate functions MUST NOT be referenced inside the *Source* expression of a **Lookup**, **LookupSet**, or **MultiLookup** function contained within an aggregate function.

### 2.341.7.25 Filtering and Aggregates

The behavior of filters and aggregate functions is specified as follows:

- Aggregates that use dataset scopes are applied after the dataset **filter** (if any) is applied to the data.
- Aggregates that use data region scopes are applied after the data region filter (if any) is applied to the data.
- Group filters are ignored for the purposes of calculating aggregates.

- The **Aggregate** aggregate function MUST NOT appear in a report that contains any [Filter](#) elements.
- **First, Last, Previous, RunningValue,** and **RowNumber** are applied after containing data region and group filters are applied.
- Aggregates that contain other aggregate functions are applied after containing data region and group filters are applied for all scopes referenced in the expression.

## 2.341.7.26 Additional Functions

This section describes additional functions that can be referenced from expressions.

### 2.341.7.26.1 Dynamic Scoping

Report items contained within a cell of a [Tablix](#) that has automatic subtotals (as a result of a drilldown) have dynamic scoping.

For example, consider a tablix that has a Year column group and a Product row group. If the value of the text box in the detail cell is =Sum(Fields!Sales.Value), each detail cell will be grouped on both year and product. However, the year subtotal (shown when the Year group is hidden) will be grouped only on product and the product subtotal will be grouped only on year (and the grand total will not be grouped on either).

The [InScope](#) function can be used to determine what the current instance is grouped on.

### 2.341.7.26.2 Function: InScope

Function	Arguments	Type	Specification
<b>InScope</b>	<i>Return</i>	<a href="#">Boolean</a>	Returns true if the current instance is within the specified scope. Otherwise, returns false.
	<i>Scope</i>	<a href="#">String</a>	Name of a dataset, group, or data region.

A typical use for the **InScope** function is to construct links to **drillthrough reports** that will work in both tablix detail cells and automatic subtotal cells, as in the following example.

```
<Drillthrough>
<ReportName>=iif(InScope("Month"),"Transactions","ProductTotByYear")</ReportName>
<Parameters>
<Parameter Name=Year>
<Value>=Fields!Year</Value>
<Omit>=Not(InScope("Year"))</Omit>
</Parameter>
<Parameter Name=Month>
<Value>=Fields!Month</Value>
<Omit>=Not(InScope("Month"))</Omit>
</Parameter>
<Parameter Name=Product>
<Value>=Fields!Product</Value>
<Omit>=Not(InScope("Product"))</Omit>
</Parameter>
</Parameters>
</Drillthrough>
```

### 2.341.7.26.3 Semantic Query Drillthrough

For report with semantic queries that use automatic drillthrough query rewriting, the function [CreateDrillthroughContext](#) is available to generate a default drillthrough context.

### 2.341.7.26.4 Function: CreateDrillthroughContext

Function	Arguments	Type	Specification
<b>CreateDrillthroughContext</b>	<i>Return</i>	<a href="#">String</a>	Returns a <b>DrillthroughContext</b> parameter value that specifies the current drillthrough context, including semantic query fields from the current scope referenced in the <b>Value</b> property of the current object (text box, image, or chart data point) and semantic query field values for the current group scopes.

This function call MUST NOT appear outside a **drillthrough parameter** value expression. Otherwise, the result is undefined.

### 2.341.7.26.5 Function: Lookup

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

Lookup functions (**Lookup**, **LookupSet**, and **MultiLookup**) provide the ability to take a value from the current scope and find a corresponding value or collection of values within a specified dataset. This is typically used in reports for converting codes into human-readable labels.

Function	Arguments	Type	Description
<b>Lookup</b>	<i>Return</i>	<b>Variant</b>	<b>Lookup</b> evaluates the <i>Source</i> expression for the current instance of the current scope. It then finds the first row of data in the specified <i>Dataset</i> * for which the source value equals the <i>Destination</i> expression evaluated for that row.** It returns the value of the <i>Result</i> expression evaluated for that row, or NULL if no matching row is found. The <i>Return</i> data type is the same as the <i>Result</i> data type.
	<i>Source</i>	<b>Variant</b>	Specifies the value to look up. The <i>Source</i> expression result MUST match the data type of the <i>Destination</i> expression. The expression MUST NOT reference lookup functions.***
	<i>Destination</i>	<b>Variant</b>	Specifies the value to compare the <i>Source</i> expression to. The <i>Destination</i> expression MUST NOT reference aggregate functions, report items, or lookup functions.
	<i>Result</i>	<b>Variant</b>	The <i>Result</i> expression MUST NOT reference aggregate functions, report items, or lookup functions.
	<i>Dataset</i>	<a href="#">String</a>	Name of the dataset in which to evaluate the <i>Destination</i> expression and <i>Result</i> expression. MUST be a constant, not an expression.

\* After *Dataset* filters have been applied.

\*\* The comparison SHOULD use collation settings from the *Dataset*.

\*\*\* "Lookup functions" refers to all types of lookup functions (such as **Lookup**, **LookupSet**, and **MultiLookup**).

### 2.341.7.26.6 Function: LookupSet

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

The main difference between [Lookup](#) and **LookupSet** is that the latter returns all matching values instead of only the first match.

Function	Arguments	Type	Description
<b>LookupSet</b>	<i>Return</i>	<b>VariantArray</b>	<p><b>LookupSet</b> evaluates the <i>Source</i> expression for the current instance of the current scope.</p> <p>It then finds all rows of data in the specified dataset for which the source value equals the <i>Destination</i> expression evaluated for that row.</p> <p>It returns an array containing a list of all values of the <i>Result</i> expression evaluated for each matching row, or an empty array if no matching rows are found. The list retains the order of the rows in the original dataset.</p> <p>Elements of the <i>Return</i> array MUST be the same data type as the <i>Result</i> data type.</p>
	<i>Source</i>	<b>Variant</b>	<p>Specifies the value to look up.</p> <p>The value of the expression result MUST match the data type of the <i>Destination</i> expression.</p> <p>The expression MUST NOT reference lookup functions.</p>
	<i>Destination</i>	<b>Variant</b>	<p>Specifies the value to compare the <i>Source</i> expression to.</p> <p>The <i>Destination</i> expression MUST NOT reference aggregate functions, report item, or lookup functions.</p>
	<i>Result</i>	<b>Variant</b>	<p>The <i>Result</i> expression MUST NOT reference aggregate functions, report items, or lookup functions.</p>
	<i>DataSet</i>	<a href="#">String (IXMLSCHEMA2/2 ↓ section 3.2.1)</a>	<p>Name of the dataset in which to evaluate the <i>Destination</i> expression and <i>Result</i> expression.</p> <p>MUST be a constant, not an expression.</p>

### 2.341.7.26.7 Function: MultiLookup

Applies to [RDL 2010/01](#) and [RDL 2016/01](#)

The following are the arguments and types of the **MultiLookup** function.

Function	Arguments	Type	Description
<b>MultiLookup</b>	<i>Return</i>	<b>VariantArray</b>	<p><b>MultiLookup</b> evaluates the <i>Source</i> expression for the current instance of the current scope.</p> <p>Then, for each item in the <i>Source</i> array, it finds the first row of data in the specified dataset for which the item in the <i>Source</i> array equals the <i>Destination</i> expression evaluated for that row. <a href="#">&lt;194&gt;</a></p> <p>It returns a <b>Variant</b> array containing a list of values of the <i>Result</i> expression evaluated for each such first matching row. <b>MultiLookup</b> returns an empty array if no matching rows are found for any item. The items in the array retain the order of the items in the <i>Source</i> array.</p> <p>Elements of the <i>Return</i> array MUST be the same data type as the <i>Result</i> type.</p>
	<i>Source</i>	<b>VariantArray</b> <a href="#">&lt;195&gt;</a>	<p>The value to look up.</p> <p>Items in the array MUST match the data type of the <i>Destination</i> expression.</p> <p>The expression MUST NOT reference lookup functions.*</p>

Function	Arguments	Type	Description
	<i>Destination</i>	<b>Variant</b>	Specifies the value to compare the <i>Source</i> expression to. The <i>Destination</i> expression MUST NOT reference aggregate functions, report items, or lookup functions.*
	<i>Result</i>	<b>Variant</b>	The expression MUST NOT reference aggregate functions, report items, or lookup functions.*
	<i>Dataset</i>	<b>String</b>	Name of the dataset in which to evaluate the <i>Destination</i> expression and <i>Result</i> expression. MUST be a constant, not an expression.

\* "Lookup functions" refers to all types of lookup functions (such as [Lookup](#), [LookupSet](#), and [MultiLookup](#)).

### 2.341.7.26.8 Function: MinValue

*Applies to* [RDL 2011/01](#)

The following are the arguments and types of the **MinValue** function.

Function	Arguments	Type	Description
<b>MinValue</b>	<i>Return</i>	<b>Variant</b>	<b>MinValue</b> accepts a variable number of arguments and returns the smallest argument value. <b>MinValue</b> MUST NOT accept less than two arguments. If any argument contains an evaluation error, <b>MinValue</b> produces an evaluation error. The data type of the <i>Return</i> value MUST be the same as the data type of the smallest argument value.
	<i>Arg1</i>	<b>Variant</b>	The first value to compare.
	<i>Arg2</i>	<b>Variant</b>	The second value to compare.
	<i>ArgN</i>	<b>Variant</b>	The Nth value to compare.

### 2.341.7.26.9 Function: MaxValue

*Applies to* [RDL 2011/01](#)

The following are the arguments and types of the **MaxValue** function.

Function	Arguments	Type	Description
<b>MaxValue</b>	<i>Return</i>	<b>Variant</b>	<b>MaxValue</b> accepts a variable number of arguments and returns the largest argument value. <b>MaxValue</b> MUST NOT accept less than two arguments. If any argument contains an evaluation error, <b>MaxValue</b> produces an evaluation error. The data type of the <i>Return</i> value MUST be the same as the data type of the smallest argument value.
	<i>Arg1</i>	<b>Variant</b>	The first value to compare.
	<i>Arg2</i>	<b>Variant</b>	The second value to compare.
	<i>ArgN</i>	<b>Variant</b>	The Nth value to compare.





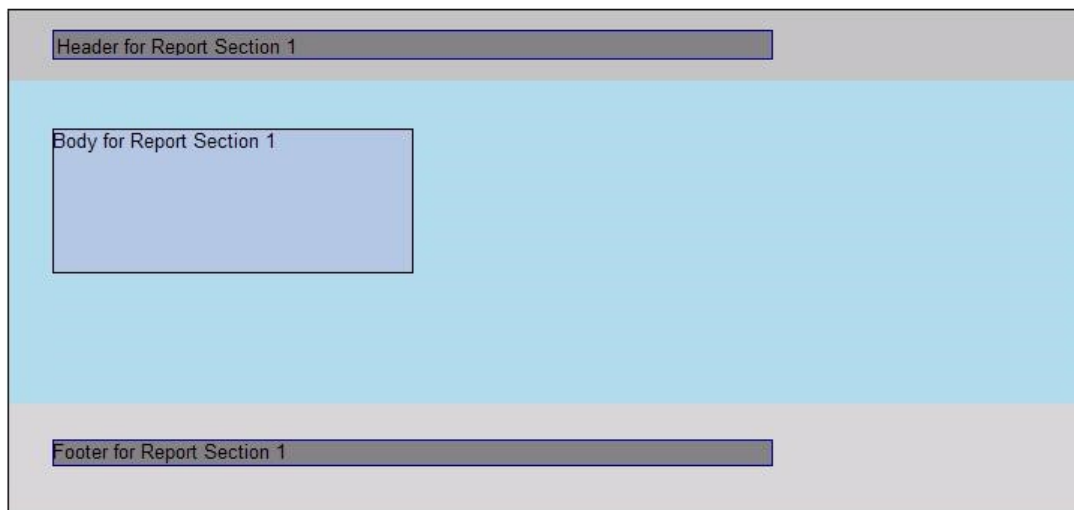
## 3 Structure Examples

The following sections provide examples for the structures defined in the RDL file format specification. Examples of the RDL file format include examples that demonstrate defining the major components of a report; these include defining the report, the data, and various report items.

### 3.1 Report in RDL schema 2008/01

The following is an example of the RDL for a report in Schema version [RDL 2008/01](#). It shows the usage of the various properties of a [Report](#). The example uses elements for describing the data: [DataSources](#) and [DataSets](#); elements for describing the structure and layout: [Page](#), including [Page.PageHeader](#) and [Page.PageFooter](#), and [Body](#); and elements for describing the [ReportParameters](#) and [Variables](#) for the report.

The following figure shows an example rendering of this report.



**Figure 9: RDL schema 2008/01 report rendering**

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <Report xmlns="http://schemas.microsoft.com/sqlserver/reporting/2008/01/reportdefinition">
3   <DataSources>
4     <DataSource Name="DataSource1">
5       <ConnectionProperties>
6         <DataProvider>SQL</DataProvider>
7         <ConnectionString>data source=DataServer; initial catalog=Northwind;</ConnectionString>
8       </ConnectionProperties>
9     </DataSource>
10  </DataSources>
11  <DataSets>
12    <DataSet Name="DataSet1">
13      <Fields>
14        <Field Name="ProductID">
15          <DataField>ProductID</DataField>
16        </Field>
17        <Field Name="ProductName">
18          <DataField>ProductName</DataField>
19        </Field>
20        <Field Name="SupplierID">
21          <DataField>SupplierID</DataField>
22        </Field>

```

```

23     <Field Name="CategoryID">
24         <DataField>CategoryID</DataField>
25     </Field>
26     <Field Name="QuantityPerUnit">
27         <DataField>QuantityPerUnit</DataField>
28     </Field>
29     <Field Name="UnitPrice">
30         <DataField>UnitPrice</DataField>
31     </Field>
32     <Field Name="UnitsInStock">
33         <DataField>UnitsInStock</DataField>
34     </Field>
35     <Field Name="UnitsOnOrder">
36         <DataField>UnitsOnOrder</DataField>
37     </Field>
38     <Field Name="ReorderLevel">
39         <DataField>ReorderLevel</DataField>
40     </Field>
41     <Field Name="Discontinued">
42         <DataField>Discontinued</DataField>
43     </Field>
44 </Fields>
45 <Query>
46     <DataSourceName>DataSourcel</DataSourceName>
47     <CommandText>Select * From Products</CommandText>
48 </Query>
49 </DataSet>
50 </DataSets>
51 <Body>
52     <ReportItems>
53         <Textbox Name="BodyTextbox">
54             <CanGrow>true</CanGrow>
55             <KeepTogether>true</KeepTogether>
56             <Paragraphs>
57                 <Paragraph>
58                     <TextRuns>
59                         <TextRun>
60                             <Value>Body</Value>
61                         </TextRun>
62                     </TextRuns>
63                 </Paragraph>
64             </Paragraphs>
65             <Top>0.33333in</Top>
66             <Left>0.3125in</Left>
67             <Height>1in</Height>
68             <Width>2.5in</Width>
69             <Style>
70                 <Border>
71                     <Color>Black</Color>
72                     <Style>Solid</Style>
73                     <Width>1pt</Width>
74                 </Border>
75                 <BackgroundColor>LightSteelBlue</BackgroundColor>
76             </Style>
77         </Textbox>
78     </ReportItems>
79     <Height>2.5in</Height>
80     <Style>
81         <BackgroundColor>LightBlue</BackgroundColor>
82     </Style>
83 </Body>
84 <Page>
85     <PageHeader>
86         <Height>0.5in</Height>
87         <PrintOnFirstPage>true</PrintOnFirstPage>
88         <PrintOnLastPage>true</PrintOnLastPage>
89     <ReportItems>
90         <Textbox Name="HeaderTextbox">
91             <Paragraphs>

```

```

92         <Paragraph>
93             <TextRuns>
94                 <TextRun>
95                     <Value>Header</Value>
96                 </TextRun>
97             </TextRuns>
98         </Paragraph>
99     </Paragraphs>
100     <Top>0.14583in</Top>
101     <Left>0.3125in</Left>
102     <Height>0.19792in</Height>
103     <Width>5in</Width>
104     <Style>
105         <Border>
106             <Color>DarkBlue</Color>
107             <Style>Solid</Style>
108             <Width>1pt</Width>
109         </Border>
110         <BackgroundColor>Gray</BackgroundColor>
111         <PaddingLeft>2pt</PaddingLeft>
112         <PaddingRight>2pt</PaddingRight>
113         <PaddingTop>2pt</PaddingTop>
114         <PaddingBottom>2pt</PaddingBottom>
115     </Style>
116 </Textbox>
117 </ReportItems>
118 <Style>
119     <BackgroundColor>Silver</BackgroundColor>
120 </Style>
121 </PageHeader>
122 <PageFooter>

123     <Height>0.75in</Height>

124 <PrintOnFirstPage>true</PrintOnFirstPage>
125 <PrintOnLastPage>true</PrintOnLastPage>
126 <ReportItems>
127     <Textbox Name="FooterTextbox">
128         <CanGrow>true</CanGrow>
129         <KeepTogether>true</KeepTogether>
130         <Paragraphs>
131             <Paragraph>
132                 <TextRuns>
133                     <TextRun>
134                         <Value>Footer</Value>
135                     </TextRun>
136                 </TextRuns>
137             </Paragraph>
138         </Paragraphs>
139         <Top>0.25in</Top>
140         <Left>0.3125in</Left>
141         <Height>0.17708in</Height>
142         <Width>5in</Width>
143         <Style>
144             <Border>
145                 <Color>DarkBlue</Color>
146                 <Style>Solid</Style>
147                 <Width>1pt</Width>
148             </Border>
149             <BackgroundColor>LightGray</BackgroundColor>
150         </Style>
151     </Textbox>
152 </ReportItems>
153 <Style>
154     <Border>
155         <Style>None</Style>
156     </Border>
157     <BackgroundColor>Silver</BackgroundColor>
158 </Style>

```

```

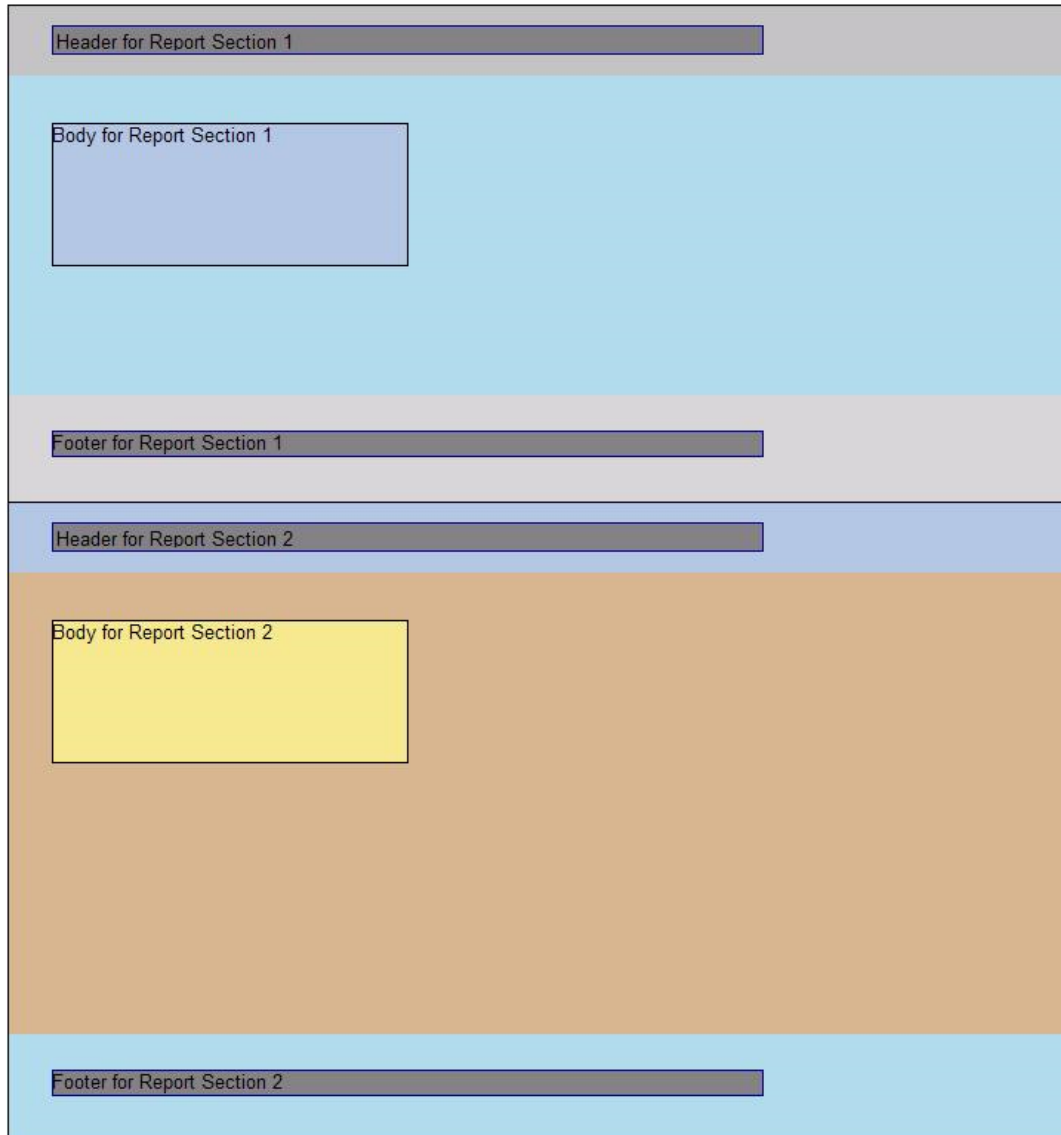
159 </PageFooter>
160 <PageWidth>8.5in</PageWidth>
161 <PageHeight>4in</PageHeight>
162 <InteractiveWidth>8.5in</InteractiveWidth>
163 <LeftMargin>0.5in</LeftMargin>
164 <RightMargin>0.5in</RightMargin>
165 <TopMargin>0.25in</TopMargin>
166 <BottomMargin>0.25in</BottomMargin>
167 <Style>
168   <Border>
169     <Color>Black</Color>
170     <Style>Solid</Style>
171     <Width>1pt</Width>
172   </Border>
173 </Style>
174 </Page>
175 <Width>6.5in</Width>
176 <Language>=User!Language</Language>
177 <Variables>
178   <Variable Name="ReportVariable1">
179     <Value>=1</Value>
180   </Variable>
181 </Variables>
182 <ConsumeContainerWhitespace>true</ConsumeContainerWhitespace>
183 <ReportParameters>
184   <ReportParameter Name="Product">
185     <DataType>String</DataType>
186     <DefaultValue>
187       <DataSetReference>
188         <DataSetName>DataSet1</DataSetName>
189         <ValueField>ProductID</ValueField>
190       </DataSetReference>
191     </DefaultValue>
192     <Prompt>Product</Prompt>
193     <ValidValues>
194       <DataSetReference>
195         <DataSetName>DataSet1</DataSetName>
196         <ValueField>ProductID</ValueField>
197         <LabelField>ProductName</LabelField>
198       </DataSetReference>
199     </ValidValues>
200     <MultiValue>true</MultiValue>
201   </ReportParameter>
202   <ReportParameter Name="Quantity">
203     <DataType>Integer</DataType>
204     <DefaultValue>
205       <Values>
206         <Value DataType="Integer">0</Value>
207       </Values>
208     </DefaultValue>
209     <Prompt>Quantity</Prompt>
210     <ValidValues>
211       <ParameterValues>
212         <ParameterValue>
213           <Value DataType="Integer">0</Value>
214           <Label>0</Label>
215         </ParameterValue>
216         <ParameterValue>
217           <Value DataType="Integer">10</Value>
218           <Label>10</Label>
219         </ParameterValue>
220         <ParameterValue>
221           <Value DataType="Integer">50</Value>
222           <Label>50</Label>
223         </ParameterValue>
224       </ParameterValues>
225     </ValidValues>
226   </ReportParameter>
227 </ReportParameters>

```

### 3.2 Report in RDL schema 2010/01

The following is an example of the RDL for a report in Schema version [RDL 2010/01](#). It shows the usage of the various properties of a [Report](#) element. The example uses elements for describing the data: [DataSources](#) and [DataSets](#); elements for describing the structure and layout for two [ReportSection](#) elements: [Page](#), including [Page.PageHeader](#) and [Page.PageFooter](#), and [Body](#); and elements for describing the [ReportParameters](#) and [Variables](#) for the report.

The following figure shows an example rendering of this report.



**Figure 10: RDL schema 2010/01 report rendering**

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <Report
3   xmlns="http://schemas.microsoft.com/sqlserver/reporting/2010/01/reportdefinition">
4   <AutoRefresh>3600</AutoRefresh>
5   <DataSources>
6     <DataSource Name="DataSource1">
7       <ConnectionProperties>
8         <DataProvider>SQL</DataProvider>
9         <ConnectionString>data source=DataServer; initial
10          catalog=northwind;</ConnectionString>
11       </ConnectionProperties>
12     </DataSource>
13   </DataSources>
14   <DataSets>
15     <DataSet Name="DataSet1">
16       <Fields>
17         <Field Name="ProductID">
18           <DataField>ProductID</DataField>
19         </Field>
20         <Field Name="ProductName">
21           <DataField>ProductName</DataField>
22         </Field>
23         <Field Name="SupplierID">
24           <DataField>SupplierID</DataField>
25         </Field>
26         <Field Name="CategoryID">
27           <DataField>CategoryID</DataField>
28         </Field>
29         <Field Name="QuantityPerUnit">
30           <DataField>QuantityPerUnit</DataField>
31         </Field>
32         <Field Name="UnitPrice">
33           <DataField>UnitPrice</DataField>
34         </Field>
35         <Field Name="UnitsInStock">
36           <DataField>UnitsInStock</DataField>
37         </Field>
38         <Field Name="UnitsOnOrder">
39           <DataField>UnitsOnOrder</DataField>
40         </Field>
41         <Field Name="ReorderLevel">
42           <DataField>ReorderLevel</DataField>
43         </Field>
44         <Field Name="Discontinued">
45           <DataField>Discontinued</DataField>
46         </Field>
47       </Fields>
48       <Query>
49         <DataSourceName>DataSource1</DataSourceName>
50         <CommandText>Select * From Products</CommandText>
51       </Query>
52     </DataSet>
53   </DataSets>
54   <ReportSections>
55     <ReportSection>
56       <Body>
57         <ReportItems>
58           <Textbox Name="BodyTextbox1">
59             <CanGrow>true</CanGrow>
60             <KeepTogether>true</KeepTogether>
61             <Paragraphs>
62               <Paragraph>
63                 <TextRuns>
64                   <TextRun>
65                     <Value>Body for Report Section 1</Value>
66                   </TextRun>
67                 </TextRuns>
68               </Paragraph>
69             </Paragraphs>

```

```

68         <Top>0.33333in</Top>
69         <Left>0.3125in</Left>
70         <Height>1in</Height>
71         <Width>2.5in</Width>
72         <Style>
73             <Border>
74                 <Color>Black</Color>
75                 <Style>Solid</Style>
76                 <Width>1pt</Width>
77             </Border>
78             <BackgroundColor>LightSteelBlue</BackgroundColor>
79         </Style>
80     </Textbox>
81 </ReportItems>
82 <Height>2.5in</Height>
83 <Style>
84     <BackgroundColor>LightBlue</BackgroundColor>
85 </Style>
86 </Body>
87 <Width>6.5in</Width>
88 <Page>
89     <PageHeader>
90         <Height>0.5in</Height>
91         <PrintOnFirstPage>true</PrintOnFirstPage>
92         <PrintOnLastPage>true</PrintOnLastPage>
93         <PrintBetweenSections>true</PrintBetweenSections>
94         <ReportItems>
95             <Textbox Name="HeaderTextbox1">
96                 <Paragraphs>
97                     <Paragraph>
98                         <TextRuns>
99                             <TextRun>
100                                 <Value>Header for Report Section 1</Value>
101                             </TextRun>
102                         </TextRuns>
103                     </Paragraph>
104                 </Paragraphs>
105                 <Top>0.14583in</Top>
106                 <Left>0.3125in</Left>
107                 <Height>0.19792in</Height>
108                 <Width>5in</Width>
109                 <Style>
110                     <Border>
111                         <Color>DarkBlue</Color>
112                         <Style>Solid</Style>
113                         <Width>1pt</Width>
114                     </Border>
115                     <BackgroundColor>Gray</BackgroundColor>
116                     <PaddingLeft>2pt</PaddingLeft>
117                     <PaddingRight>2pt</PaddingRight>
118                     <PaddingTop>2pt</PaddingTop>
119                     <PaddingBottom>2pt</PaddingBottom>
120                 </Style>
121             </Textbox>
122         </ReportItems>
123     <Style>
124         <BackgroundColor>Silver</BackgroundColor>
125     </Style>
126 </PageHeader>
127 <PageFooter>
128     <Height>0.75in</Height>
129     <PrintOnFirstPage>true</PrintOnFirstPage>
130     <PrintOnLastPage>true</PrintOnLastPage>
131     <PrintBetweenSections>true</PrintBetweenSections>
132     <ReportItems>
133         <Textbox Name="FooterTextbox1">
134             <CanGrow>true</CanGrow>
135             <KeepTogether>true</KeepTogether>
136             <Paragraphs>

```



```

137         <Paragraph>
138             <TextRuns>
139                 <TextRun>
140                     <Value>Footer for Report Section 1</Value>
141                 </TextRun>
142             </TextRuns>
143         </Paragraph>
144     </Paragraphs>
145     <Top>0.25in</Top>
146     <Left>0.3125in</Left>
147     <Height>0.17708in</Height>
148     <Width>5in</Width>
149     <Style>
150         <Border>
151             <Color>DarkBlue</Color>
152             <Style>Solid</Style>
153             <Width>1pt</Width>
154         </Border>
155         <BackgroundColor>Gray</BackgroundColor>
156     </Style>
157 </Textbox>
158 </ReportItems>
159 <Style>
160     <BackgroundColor>LightGrey</BackgroundColor>
161 </Style>
162 </PageFooter>
163 <PageWidth>8.5in</PageWidth>
164 <PageHeight>5in</PageHeight>
165 <InteractiveWidth>8.5in</InteractiveWidth>
166 <LeftMargin>0.5in</LeftMargin>
167 <RightMargin>0.5in</RightMargin>
168 <TopMargin>0.25in</TopMargin>
169 <BottomMargin>0.25in</BottomMargin>
170 <Style>
171     <Border>
172         <Color>Black</Color>
173         <Style>Solid</Style>
174         <Width>1pt</Width>
175     </Border>
176 </Style>
177 </Page>
178 </ReportSection>
179 <ReportSection>
180     <Body>
181         <ReportItems>
182             <Textbox Name="BodyTextbox2">
183                 <CanGrow>true</CanGrow>
184                 <KeepTogether>true</KeepTogether>
185                 <Paragraphs>
186                     <Paragraph>
187                         <TextRuns>
188                             <TextRun>
189                                 <Value>Body for Report Section 2</Value>
190                             </TextRun>
191                         </TextRuns>
192                     </Paragraph>
193                 </Paragraphs>
194                 <Top>0.33333in</Top>
195                 <Left>0.3125in</Left>
196                 <Height>1in</Height>
197                 <Width>2.5in</Width>
198                 <Style>
199                     <Border>
200                         <Color>Black</Color>
201                         <Style>Solid</Style>
202                         <Width>1pt</Width>
203                     </Border>
204                     <BackgroundColor>Khaki</BackgroundColor>
205                 </Style>

```

```

206     </Textbox>
207 </ReportItems>
208 <Height>3in</Height>
209 <Style>
210   <BackgroundColor>Tan</BackgroundColor>
211 </Style>
212 </Body>
213 <Width>6.5in</Width>
214 <Page>
215   <PageHeader>
216     <Height>0.5in</Height>
217     <PrintOnFirstPage>true</PrintOnFirstPage>
218     <PrintOnLastPage>true</PrintOnLastPage>
219     <PrintBetweenSections>true</PrintBetweenSections>
220     <ReportItems>
221       <Textbox Name="HeaderTextbox2">
222         <Paragraphs>
223           <Paragraph>
224             <TextRuns>
225               <TextRun>
226                 <Value>Header for Report Section 2</Value>
227               </TextRun>
228             </TextRuns>
229           </Paragraph>
230         </Paragraphs>
231         <Top>0.14583in</Top>
232         <Left>0.3125in</Left>
233         <Height>0.19792in</Height>
234         <Width>5in</Width>
235         <Style>
236           <Border>
237             <Color>DarkBlue</Color>
238             <Style>Solid</Style>
239             <Width>1pt</Width>
240           </Border>
241           <BackgroundColor>Gray</BackgroundColor>
242           <PaddingLeft>2pt</PaddingLeft>
243           <PaddingRight>2pt</PaddingRight>
244           <PaddingTop>2pt</PaddingTop>
245           <PaddingBottom>2pt</PaddingBottom>
246         </Style>
247       </Textbox>
248     </ReportItems>
249     <Style>
250       <BackgroundColor>LightSteelBlue</BackgroundColor>
251     </Style>
252   </PageHeader>
253   <PageFooter>
254     <Height>0.75in</Height>
255     <PrintOnFirstPage>true</PrintOnFirstPage>
256     <PrintOnLastPage>true</PrintOnLastPage>
257     <PrintBetweenSections>true</PrintBetweenSections>
258     <ReportItems>
259       <Textbox Name="FooterTextbox2">
260         <CanGrow>true</CanGrow>
261         <KeepTogether>true</KeepTogether>
262         <Paragraphs>
263           <Paragraph>
264             <TextRuns>
265               <TextRun>
266                 <Value>Footer for Report Section 2</Value>
267               </TextRun>
268             </TextRuns>
269           </Paragraph>
270         </Paragraphs>
271         <Top>0.25in</Top>
272         <Left>0.3125in</Left>
273         <Height>0.17708in</Height>
274         <Width>5in</Width>

```

```

275         <Style>
276             <Border>
277                 <Color>DarkBlue</Color>
278                 <Style>Solid</Style>
279                 <Width>1pt</Width>
280             </Border>
281             <BackgroundColor>Gray</BackgroundColor>
282         </Style>
283     </Textbox>
284 </ReportItems>
285     <Style>
286         <BackgroundColor>LightBlue</BackgroundColor>
287     </Style>
288 </PageFooter>
289 <PageWidth>8.5in</PageWidth>
290 <PageHeight>5in</PageHeight>
291 <InteractiveWidth>8.5in</InteractiveWidth>
292 <LeftMargin>0.5in</LeftMargin>
293 <RightMargin>0.5in</RightMargin>
294 <TopMargin>0.25in</TopMargin>
295 <BottomMargin>0.25in</BottomMargin>
296 <Style>
297     <Border>
298         <Color>Black</Color>
299         <Style>Solid</Style>
300         <Width>1pt</Width>
301     </Border>
302 </Style>
303 </Page>
304 </ReportSection>
305 </ReportSections>
306 <Language>=User!Language</Language>
307 <Variables>
308     <Variable Name="ReportVariable1">
309         <Value>=1</Value>
310     </Variable>
311 </Variables>
312 <ConsumeContainerWhitespace>true</ConsumeContainerWhitespace>
313 <ReportParameters>
314     <ReportParameter Name="Product">
315         <DataType>String</DataType>
316         <DefaultValue>
317             <DataSetReference>
318                 <DataSetName>DataSet1</DataSetName>
319                 <ValueField>ProductID</ValueField>
320             </DataSetReference>
321         </DefaultValue>
322         <Prompt>Product</Prompt>
323         <ValidValues>
324             <DataSetReference>
325                 <DataSetName>DataSet1</DataSetName>
326                 <ValueField>ProductID</ValueField>
327                 <LabelField>ProductName</LabelField>
328             </DataSetReference>
329         </ValidValues>
330         <MultiValue>true</MultiValue>
331     </ReportParameter>
332     <ReportParameter Name="Quantity">
333         <DataType>Integer</DataType>
334         <DefaultValue>
335             <Values>
336                 <Value DataType="Integer">0</Value>
337             </Values>
338         </DefaultValue>
339         <Prompt>Quantity</Prompt>
340         <ValidValues>
341             <ParameterValues>
342                 <ParameterValue>
343                     <Value DataType="Integer">0</Value>

```

```

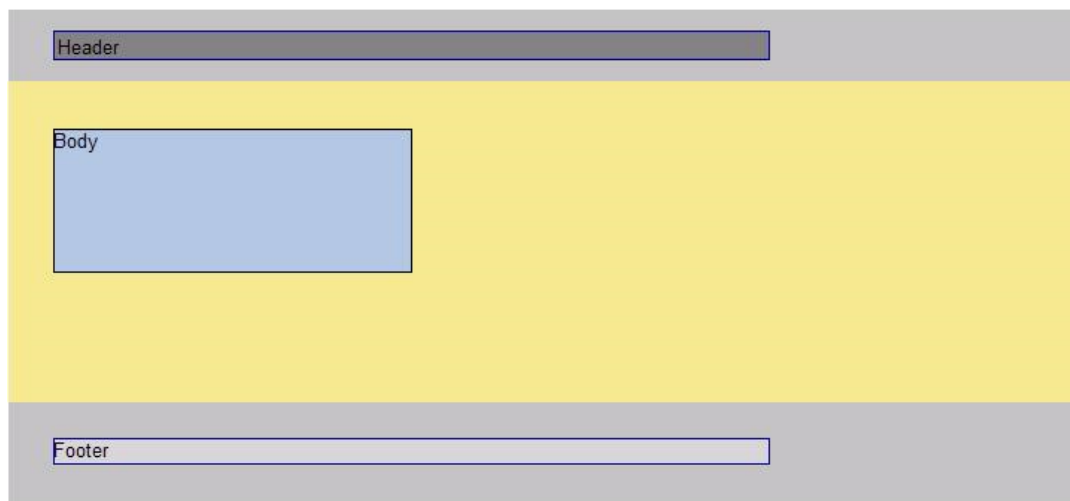
344         <Label>0</Label>
345     </ParameterValue>
346     <ParameterValue>
347         <Value DataType="Integer">10</Value>
348         <Label>10</Label>
349     </ParameterValue>
350     <ParameterValue>
351         <Value DataType="Integer">50</Value>
352         <Label>50</Label>
353     </ParameterValue>
354 </ParameterValues>
355 </ValidValues>
356 </ReportParameter>
357 </ReportParameters>
358</Report>

```

### 3.3 Report in RDL schema 2005/01

The following is an example of the RDL for a report in Schema version [RDL 2005/01](#). It shows the usage of the various properties of a [Report](#). This example shows elements for describing the data: [DataSources](#) and [DataSets](#); elements for describing the structure and layout: [Report.PageHeader](#), [Report.PageFooter](#), and [Body](#); and elements for describing the [ReportParameters](#) for the report.

The following figure shows an example rendering of this report.



**Figure 11: RDL schema 2005/01 report rendering**

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <Report xmlns="http://schemas.microsoft.com/sqlserver/reporting/2005/01/reportdefinition">
3   <DataSources>
4     <DataSource Name="DataSource1">
5       <ConnectionProperties>
6         <DataProvider>SQL</DataProvider>
7         <ConnectionString>data source=DataServer; initial catalog=Northwind;</ConnectionString>
8       </ConnectionProperties>
9     </DataSource>
10  </DataSources>
11  <DataSets>
12    <DataSet Name="DataSet1">
13      <Fields>

```

```

14     <Field Name="ProductID">
15         <DataField>ProductID</DataField>
16     </Field>
17     <Field Name="ProductName">
18         <DataField>ProductName</DataField>
19     </Field>
20     <Field Name="SupplierID">
21         <DataField>SupplierID</DataField>
22     </Field>
23     <Field Name="CategoryID">
24         <DataField>CategoryID</DataField>
25     </Field>
26     <Field Name="QuantityPerUnit">
27         <DataField>QuantityPerUnit</DataField>
28     </Field>
29     <Field Name="UnitPrice">
30         <DataField>UnitPrice</DataField>
31     </Field>
32     <Field Name="UnitsInStock">
33         <DataField>UnitsInStock</DataField>
34     </Field>
35     <Field Name="UnitsOnOrder">
36         <DataField>UnitsOnOrder</DataField>
37     </Field>
38     <Field Name="ReorderLevel">
39         <DataField>ReorderLevel</DataField>
40     </Field>
41     <Field Name="Discontinued">
42         <DataField>Discontinued</DataField>
43     </Field>
44 </Fields>
45 <Query>
46     <DataSourceName>DataSource1</DataSourceName>
47     <CommandText>Select * From Products</CommandText>
48 </Query>
49 </DataSet>
50 </DataSets>
51 <Body>
52     <ReportItems>
53         <Textbox Name="BodyTextbox">
54             <CanGrow>true</CanGrow>
55             <Value>Body</Value>
56             <Top>0.33333in</Top>
57             <Left>0.3125in</Left>
58             <Height>1in</Height>
59             <Width>2.5in</Width>
60             <Style>
61                 <BorderColor>
62                     <Default>Black</Default>
63                 </BorderColor>
64                 <BorderStyle>
65                     <Default>Solid</Default>
66                 </BorderStyle>
67                 <BorderWidth>
68                     <Default>1pt</Default>
69                 </BorderWidth>
70                 <BackgroundColor>LightSteelBlue</BackgroundColor>
71             </Style>
72         </Textbox>
73     </ReportItems>
74 <Height>2.5in</Height>
75 <Style>
76     <BackgroundColor>Khaki</BackgroundColor>
77 </Style>
78 </Body>
79 <PageHeader>
80     <Height>0.5in</Height>
81     <PrintOnFirstPage>true</PrintOnFirstPage>
82     <PrintOnLastPage>true</PrintOnLastPage>

```

```

83 <ReportItems>
84   <Textbox Name="HeaderTextbox">
85     <Value>Header</Value>
86     <Top>0.14583in</Top>
87     <Left>0.3125in</Left>
88     <Height>0.19792in</Height>
89     <Width>5in</Width>
90     <Style>
91       <BorderColor>
92         <Default>DarkBlue</Default>
93       </BorderColor>
94       <BorderStyle>
95         <Default>Solid</Default>
96       </BorderStyle>
97       <BorderWidth>
98         <Default>1pt</Default>
99       </BorderWidth>
100      <BackgroundColor>Gray</BackgroundColor>
101      <PaddingLeft>2pt</PaddingLeft>
102      <PaddingRight>2pt</PaddingRight>
103      <PaddingTop>2pt</PaddingTop>
104      <PaddingBottom>2pt</PaddingBottom>
105    </Style>
106  </Textbox>
107 </ReportItems>
108 <Style>
109   <BackgroundColor>Silver</BackgroundColor>
110 </Style>
111 </PageHeader>
112 <PageFooter>
113   <Height>0.75in</Height>
114   <PrintOnFirstPage>true</PrintOnFirstPage>
115   <PrintOnLastPage>true</PrintOnLastPage>
116   <ReportItems>
117     <Textbox Name="FooterTextbox">
118       <CanGrow>true</CanGrow>
119       <Value>Footer</Value>
120       <Top>0.25in</Top>
121       <Left>0.3125in</Left>
122       <Height>0.17708in</Height>
123       <Width>5in</Width>
124       <Style>
125         <BorderColor>
126           <Default>DarkBlue</Default>
127         </BorderColor>
128         <BorderStyle>
129           <Default>Solid</Default>
130         </BorderStyle>
131         <BorderWidth>
132           <Default>1pt</Default>
133         </BorderWidth>
134         <BackgroundColor>LightGray</BackgroundColor>
135       </Style>
136     </Textbox>
137   </ReportItems>
138   <Style>
139     <BackgroundColor>Silver</BackgroundColor>
140   </Style>
141 </PageFooter>
142 <PageWidth>8.5in</PageWidth>
143 <PageHeight>6in</PageHeight>
144 <InteractiveWidth>8.5in</InteractiveWidth>
145 <LeftMargin>0.5in</LeftMargin>
146 <RightMargin>0.5in</RightMargin>
147 <TopMargin>0.25in</TopMargin>
148 <BottomMargin>0.25in</BottomMargin>
149 <Width>6.5in</Width>
150 <Language>=User!Language</Language>
151 <ReportParameters>

```

```

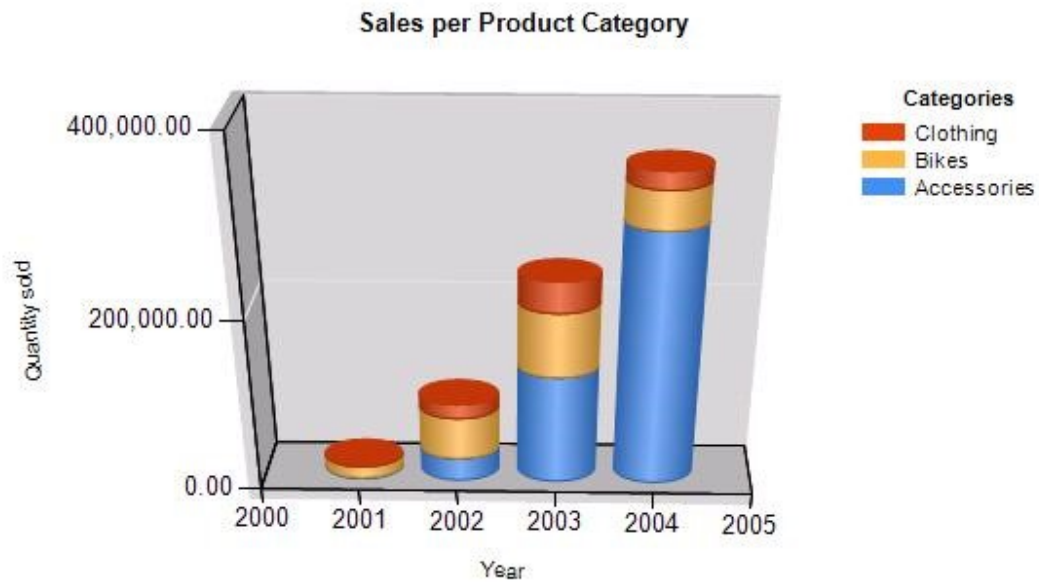
152 <ReportParameter Name="Product">
153   <DataType>String</DataType>
154   <DefaultValue>
155     <DataSetReference>
156       <DataSetName>DataSet1</DataSetName>
157       <ValueField>ProductID</ValueField>
158     </DataSetReference>
159   </DefaultValue>
160   <Prompt>Product</Prompt>
161   <ValidValues>
162     <DataSetReference>
163       <DataSetName>DataSet1</DataSetName>
164       <ValueField>ProductID</ValueField>
165       <LabelField>ProductName</LabelField>
166     </DataSetReference>
167   </ValidValues>
168   <MultiValue>>true</MultiValue>
169 </ReportParameter>
170 <ReportParameter Name="Quantity">
171   <DataType>Integer</DataType>
172   <DefaultValue>
173     <Values>
174       <Value>0</Value>
175     </Values>
176   </DefaultValue>
177   <Prompt>Quantity</Prompt>
178   <ValidValues>
179     <ParameterValues>
180       <ParameterValue>
181         <Value>0</Value>
182         <Label>0</Label>
183       </ParameterValue>
184       <ParameterValue>
185         <Value>10</Value>
186         <Label>10</Label>
187       </ParameterValue>
188       <ParameterValue>
189         <Value>50</Value>
190         <Label>50</Label>
191       </ParameterValue>
192     </ParameterValues>
193   </ValidValues>
194 </ReportParameter>
195 </ReportParameters>
196</Report>

```

### 3.4 Column and Line Chart

**Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)**

The following example report shows two independent charts. The first [Chart](#)—referenced as Chart1—starts from line 90 to line 379 in the RDL example at the end of this topic.



**Figure 12: Chart1 example**

This is a column chart with one set of series and series grouping. To understand how this chart is bound to data, consider the following table that shows the data that this chart displays.

Product category name	Year	Quantity
Accessories	2001	1003
	2002	27207
	2003	23734
	2003	103978
	2004	292660
Bikes	2001	14256
	2002	49810
	2003	30773
	2003	43241
	2004	42456
Clothing	2001	2132
	2002	16927
	2003	16515
	2003	18862
	2004	19234
Components	2001	75892



Product category name	Year	Quantity
	2002	332885
	2003	266860
	2003	314930
	2004	661065

The [Query](#) and the data fields are specified from line 10 to line 44. **Chart** is bound to the [DataSet](#) "SalesPerProduct" (see line 366).

The category axis displays the value of the Year column, the value axis displays the value of the Quantity column, and the data values are grouped by the Product category name column.

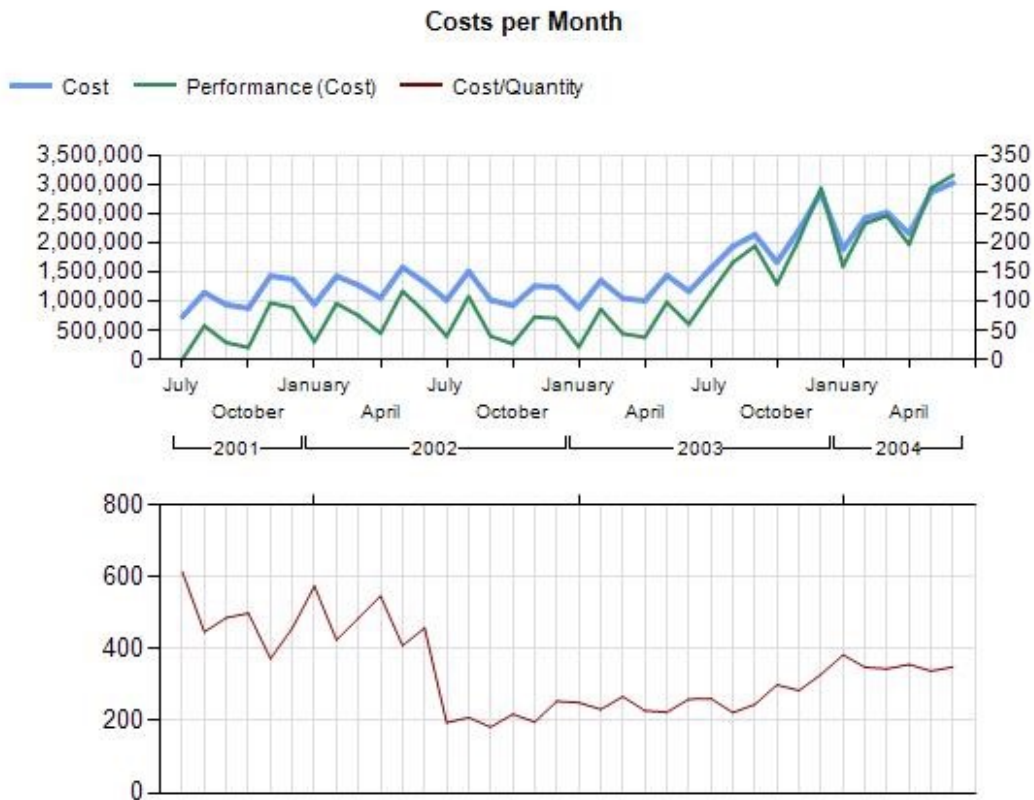
The category hierarchy is specified from line 91 to line 102; the series hierarchy is specified from line 103 to line 123.

Note that the last product category in the table is not displayed in the chart. There is a **filter** applied to the series grouping (lines 110 to 118) to skip every row of the **dataset** where the product category name equals to "Components".

This chart has only one set of series specified (its name is "Quantity") in the [ChartSeriesCollection](#) element (lines 125 to 167).

Because this is a 3D chart with the projection mode set to "Perspective", 3D properties are specified from line 308 to line 313, except for the shape of the columns, which is specified by custom properties for the series (from lines 154 to 159).

The second chart (specified as Chart2 from line 380 to line 898) displays three series (one of them is calculated) with nested category grouping. This example also shows how to use secondary axes and how to align chart areas to match gridlines.



**Figure 13: Chart2 example**

Chart2 has two chart areas:

- Chart area "Default" (lines 533 to 676) displays series (*Cost*) and calculated series (*Performance (Cost)*).
- "ChartArea2" (lines 677 to 835) displays the *Cost/Quantity* series; it is aligned to chart area one on the inner plot position (line 830 to line 834).

Note that on the first chart area, the category axis displays a hierarchy of years and months. To understand this behavior, consider the following table that shows a subset of data applied to the chart. (The **query** and the data fields are specified from line 45 to line 87. The **Chart** is bound to the **DataSet** "CostsPerMonth".)

Year	Month	Cost	Quantity
2001	7	726727.4476	1187
	8	1142665.9223	2560
	9	934633.6329	1924
	10	873968.9087	1754
	11	1427743.5228	3837
	12	1369873.0300	2994

Year	Month	Cost	Quantity
2002	1	945210.0341	1649
	2	1421680.2332	3355
	3	1270261.5659	2620
	4	1051384.7585	1927
	5	1573531.6562	3854
	6	1318811.3400	2884
	7	1013621.4783	5226
	8	1506516.6655	7227
	9	1014436.2328	5582
	10	920017.4347	4223
	11	1253971.9866	6396
	12	1235027.0052	4867

This table has two columns that are used for category grouping. Look at the embedded category **members** from line 381 to line 402. Also notice that—although this chart has no series grouping—two static members are specified in the series hierarchy (lines 403 to 412).

The hierarchy of category groups is automatically reflected on the category axis: Year is the parent group, and Month is the child group; therefore, **Axis** does not need any alteration.

As previously mentioned, the first chart area displays two series. The second set of series is calculated, which means that it takes the values of the first series and, depending on the formula that is applied, displays its values as a different series.

The calculated series (Series1) is specified from line 492 to line 530. Note that calculated series need references to the series they're applied to (line 527). Because the values of the calculated series are in a different range, the secondary axis is used. To achieve this, it is sufficient to assign the series to the secondary value axis (line 520) as long as the [Visible](#) property of the related axis is set to "Auto" (because "Auto" is the default value for the **Visible** element, it is not present in the RDL file).

The second area shows two category axes. Because this chart area is aligned to the previous one and the chart displays values in the same range, it is not necessary to show anything on any of the axes. Instead, the primary axis of the first chart area is reused.

The primary axis is enabled to display a frame at the bottom of the chart area, but it does not display labels (see line 715).

The secondary axis, however, is visible (line 718), and the [ChartMajorTickMarks](#) elements' [Interval](#) and [IntervalOffset](#) properties are used to display tick marks for the first month for each year (from lines 744 to 748).

```

1<?xml version="1.0" encoding="utf-8"?>
2<Report xmlns:rd="http://schemas.microsoft.com/SQLServer/reporting/reportdesigner"
  xmlns="http://schemas.microsoft.com/sqlserver/reporting/2008/01/reportdefinition">
3  <DataSources>
4    <DataSource Name="Chart">
5      <DataSourceReference>AdventureWorks</DataSourceReference>
6      <rd:DataSourceID>98587803-9a41-40bd-9855-5bf01cb26c13</rd:DataSourceID>
7    </DataSource>

```

```

8  </DataSources>
9  <DataSets>
10   <DataSet Name="SalesPerProduct">
11     <Fields>
12       <Field Name="ProductCategoryName">
13         <DataField>ProductCategoryName</DataField>
14         <rd:TypeName>System.String</rd:TypeName>
15       </Field>
16       <Field Name="Year">
17         <DataField>Year</DataField>
18         <rd:TypeName>System.Int32</rd:TypeName>
19       </Field>
20       <Field Name="Quantity">
21         <DataField>Quantity</DataField>
22         <rd:TypeName>System.Int32</rd:TypeName>
23       </Field>
24     </Fields>
25     <Query>
26       <DataSourceName>Chart</DataSourceName>
27       <CommandText>SELECT      Production.ProductCategory.Name AS ProductCategoryName,
YEAR(Production.TransactionHistory.TransactionDate) AS Year,
28         SUM(Production.TransactionHistory.Quantity) AS Quantity
29FROM      Production.Product INNER JOIN
30         Production.ProductSubcategory ON
Production.Product.ProductSubcategoryID = Production.ProductSubcategory.ProductSubcategoryID
INNER JOIN
31         Production.ProductCategory ON
Production.ProductSubcategory.ProductCategoryID =
Production.ProductCategory.ProductCategoryID INNER JOIN
32         Production.TransactionHistory ON Production.Product.ProductID =
Production.TransactionHistory.ProductID
33GROUP BY Production.ProductCategory.Name,
YEAR(Production.TransactionHistory.TransactionDate)
34UNION
35SELECT      Production.ProductCategory.Name AS ProductCategoryName,
YEAR(Production.TransactionHistoryArchive.TransactionDate) AS Year,
36         SUM(Production.TransactionHistoryArchive.Quantity) AS Quantity
37FROM      Production.Product INNER JOIN
38         Production.ProductSubcategory ON
Production.Product.ProductSubcategoryID = Production.ProductSubcategory.ProductSubcategoryID
INNER JOIN
39         Production.ProductCategory ON
Production.ProductSubcategory.ProductCategoryID =
Production.ProductCategory.ProductCategoryID INNER JOIN
40         Production.TransactionHistoryArchive ON Production.Product.ProductID
= Production.TransactionHistoryArchive.ProductID
41GROUP BY Production.ProductCategory.Name,
YEAR(Production.TransactionHistoryArchive.TransactionDate)</CommandText>
42       <rd:UseGenericDesigner>true</rd:UseGenericDesigner>
43     </Query>
44   </DataSet>
45   <DataSet Name="CostsPerMonth">
46     <Fields>
47       <Field Name="Year">
48         <DataField>Year</DataField>
49         <rd:TypeName>System.Int32</rd:TypeName>
50       </Field>
51       <Field Name="Month">
52         <DataField>Month</DataField>
53         <rd:TypeName>System.Int32</rd:TypeName>
54       </Field>
55       <Field Name="Cost">
56         <DataField>Cost</DataField>
57         <rd:TypeName>System.Decimal</rd:TypeName>
58       </Field>
59       <Field Name="Quantity">
60         <DataField>Quantity</DataField>
61         <rd:TypeName>System.Int32</rd:TypeName>
62       </Field>

```

```

63     </Fields>
64     <Query>
65         <DataSourceName>Chart</DataSourceName>
66         <CommandText>SELECT      YEAR(Production.TransactionHistory.TransactionDate) AS
Year, MONTH(Production.TransactionHistory.TransactionDate) AS Month,
67             SUM(Production.TransactionHistory.ActualCost) AS Cost,
SUM(Production.TransactionHistory.Quantity) AS Quantity
68FROM      Production. Product INNER JOIN
69             Production.ProductSubcategory ON Production. Product
.ProductSubcategoryID = Production.ProductSubcategory.ProductSubcategoryID INNER JOIN
70             Production.ProductCategory ON
Production.ProductSubcategory.ProductCategoryID =
Production.ProductCategory.ProductCategoryID INNER JOIN
71             Production.TransactionHistory ON Production. Product .ProductID =
Production.TransactionHistory.ProductID
72WHERE     (Production.ProductCategory.Name = 'Bikes')
73GROUP BY YEAR(Production.TransactionHistory.TransactionDate),
MONTH(Production.TransactionHistory.TransactionDate)
74HAVING    SUM(Production.TransactionHistory.ActualCost) > 0
75UNION ALL
76SELECT    YEAR(Production.TransactionHistoryArchive.TransactionDate) AS Year,
MONTH(Production.TransactionHistoryArchive.TransactionDate) AS Month,
77             SUM(Production.TransactionHistoryArchive.ActualCost) AS Cost,
SUM(Production.TransactionHistoryArchive.Quantity) AS Quantity
78FROM      Production. Product INNER JOIN
79             Production.ProductSubcategory ON Production. Product
.ProductSubcategoryID = Production.ProductSubcategory.ProductSubcategoryID INNER JOIN
80             Production.ProductCategory ON
Production.ProductSubcategory.ProductCategoryID =
Production.ProductCategory.ProductCategoryID INNER JOIN
81             Production.TransactionHistoryArchive ON Production. Product
.ProductID = Production.TransactionHistoryArchive.ProductID
82WHERE     (Production.ProductCategory.Name = 'Bikes')
83GROUP BY YEAR(Production.TransactionHistoryArchive.TransactionDate),
MONTH(Production.TransactionHistoryArchive.TransactionDate)
84ORDER BY Year, Month</CommandText>
85     </Query>
86 </DataSet>
87 </DataSets>
88 <Body>
89     <ReportItems>
90         <Chart Name="Chart1">
91             <ChartCategoryHierarchy>
92                 <ChartMembers>
93                     <ChartMember>
94                         <Group Name="Chart1 CategoryGroup1">
95                             <GroupExpressions>
96                                 <GroupExpression>=Fields!Year.Value</GroupExpression>
97                             </GroupExpressions>
98                         </Group>
99                         <Label>=Fields!Year.Value</Label>
100                    </ChartMember>
101                </ChartMembers>
102            </ChartCategoryHierarchy>
103            <ChartSeriesHierarchy>
104                <ChartMembers>
105                    <ChartMember>
106                        <Group Name="Chart1 SeriesGroup1">
107                            <GroupExpressions>
108                                <GroupExpression>=Fields!ProductCategoryName.Value</GroupExpression>
109                            </GroupExpressions>
110                        </ChartMember>
111                    </ChartMembers>
112                </ChartSeriesHierarchy>
113            </ChartSeriesHierarchy>
114            <Filters>
115                <Filter>
116                    <FilterExpression>=Fields!ProductCategoryName.Value</FilterExpression>
</Filter>
</Filters>
</Chart>
</ReportItems>
</Body>

```

```

117         </Filter>
118     </Filters>
119 </Group>
120     <Label>=Fields!ProductCategoryName.Value</Label>
121 </ChartMember>
122 </ChartMembers>
123 </ChartSeriesHierarchy>
124 <ChartData>
125     <ChartSeriesCollection>
126         <ChartSeries Name="Quantity">
127             <ChartDataPoints>
128                 <ChartDataPoint>
129                     <ChartDataPointValues>
130                         <X>=Fields!Year.Value</X>
131                         <Y>=Sum(Fields!Quantity.Value)</Y>
132                     </ChartDataPointValues>
133                     <ChartDataLabel>
134                         <Style />
135                     </ChartDataLabel>
136                     <Style />
137                     <ChartMarker>
138                         <Style />
139                     </ChartMarker>
140                     <DataElementOutput>Output</DataElementOutput>
141                 </ChartDataPoint>
142             </ChartDataPoints>
143             <Subtype>Stacked</Subtype>
144             <Style />
145             <ChartEmptyPoints>
146                 <Style />
147                 <ChartMarker>
148                     <Style />
149                 </ChartMarker>
150                 <ChartDataLabel>
151                     <Style />
152                 </ChartDataLabel>
153             </ChartEmptyPoints>
154             <CustomProperties>
155                 <CustomProperty>
156                     <Name>DrawingStyle</Name>
157                     <Value>Cylinder</Value>
158                 </CustomProperty>
159             </CustomProperties>
160             <ValueAxisName>Primary</ValueAxisName>
161             <CategoryAxisName>Primary</CategoryAxisName>
162             <ChartSmartLabel>
163                 <CalloutLineColor>Black</CalloutLineColor>
164                 <MinMovingDistance>0pt</MinMovingDistance>
165             </ChartSmartLabel>
166         </ChartSeries>
167     </ChartSeriesCollection>
168 </ChartData>
169 <ChartAreas>
170     <ChartArea Name="Default">
171         <ChartCategoryAxes>
172             <ChartAxis Name="Primary">
173                 <Style>
174                     <FontSize>8pt</FontSize>
175                 </Style>
176                 <ChartAxisTitle>
177                     <Caption>Year</Caption>
178                     <Style>
179                         <FontSize>8pt</FontSize>
180                     </Style>
181                 </ChartAxisTitle>
182                 <ChartMajorGridLines>
183                     <Enabled>False</Enabled>
184                 <Style>
185                     <Border>

```

```

186         <Color>Gainsboro</Color>
187     </Border>
188 </Style>
189 </ChartMajorGridLines>
190 <ChartMinorGridLines>
191     <Style>
192         <Border>
193             <Color>Gainsboro</Color>
194             <Style>Dotted</Style>
195         </Border>
196     </Style>
197 </ChartMinorGridLines>
198 <ChartMinorTickMarks>
199     <Length>0.5</Length>
200 </ChartMinorTickMarks>

201     <CrossAt>NaN</CrossAt>
202     <Minimum>NaN</Minimum>
203     <Maximum>NaN</Maximum>
204 </ChartAxis>
205 <ChartAxis Name="Secondary">
206     <Style>
207         <FontSize>8pt</FontSize>
208     </Style>
209     <ChartAxisTitle>
210         <Caption>Axis Title</Caption>
211     </Style>
212         <FontSize>8pt</FontSize>
213     </Style>
214 </ChartAxisTitle>
215 <ChartMajorGridLines>
216     <Enabled>False</Enabled>
217 <Style>
218     <Border>
219         <Color>Gainsboro</Color>
220     </Border>
221 </Style>
222 </ChartMajorGridLines>
223 <ChartMinorGridLines>
224     <Style>
225         <Border>
226             <Color>Gainsboro</Color>
227             <Style>Dotted</Style>
228         </Border>
229     </Style>
230 </ChartMinorGridLines>
231 <ChartMinorTickMarks>
232     <Length>0.5</Length>
233 </ChartMinorTickMarks>
234 <CrossAt>NaN</CrossAt>
235 <Location>Opposite</Location>
236 <Minimum>NaN</Minimum>
237 <Maximum>NaN</Maximum>
238 </ChartAxis>
239 </ChartCategoryAxes>
240 <ChartValueAxes>
241     <ChartAxis Name="Primary">
242         <Style>
243             <FontSize>8pt</FontSize>
244             <Format>#,0.00;(#,0.00)</Format>
245         </Style>
246         <ChartAxisTitle>
247             <Caption>Quantity sold</Caption>
248         </Style>
249             <FontSize>8pt</FontSize>
250         </Style>
251     </ChartAxisTitle>
252 <ChartMajorGridLines>
253     <Style>

```

```

254         <Border>
255             <Color>Gainsboro</Color>
256         </Border>
257     </Style>
258 </ChartMajorGridLines>
259 <ChartMinorGridLines>
260     <Style>
261         <Border>
262             <Color>Gainsboro</Color>
263             <Style>Dotted</Style>
264         </Border>
265     </Style>
266 </ChartMinorGridLines>
267 <ChartMinorTickMarks>
268     <Length>0.5</Length>
269 </ChartMinorTickMarks>
270 <CrossAt>NaN</CrossAt>
271 <Minimum>NaN</Minimum>
272 <Maximum>NaN</Maximum>
273 </ChartAxis>
274 <ChartAxis Name="Secondary">
275     <Style>
276         <FontSize>8pt</FontSize>
277     </Style>
278     <ChartAxisTitle>
279         <Caption>Axis Title</Caption>
280     </Style>
281         <FontSize>8pt</FontSize>
282     </Style>
283 </ChartAxisTitle>
284 <ChartMajorGridLines>
285     <Style>
286         <Border>
287             <Color>Gainsboro</Color>
288         </Border>
289     </Style>
290 </ChartMajorGridLines>
291 <ChartMinorGridLines>
292     <Style>
293         <Border>
294             <Color>Gainsboro</Color>
295             <Style>Dotted</Style>
296         </Border>
297     </Style>
298 </ChartMinorGridLines>
299 <ChartMinorTickMarks>
300     <Length>0.5</Length>
301 </ChartMinorTickMarks>
302 <CrossAt>NaN</CrossAt>
303 <Location>Opposite</Location>
304 <Minimum>NaN</Minimum>
305 <Maximum>NaN</Maximum>
306 </ChartAxis>
307 </ChartValueAxes>
308 <ChartThreeDProperties>
309     <Enabled>true</Enabled>
310     <ProjectionMode>Perspective</ProjectionMode>
311     <Perspective>9</Perspective>
312     <Rotation>2</Rotation>
313 </ChartThreeDProperties>
314 <Style>
315     <BackgroundGradientType>None</BackgroundGradientType>
316 </Style>
317 </ChartArea>
318 </ChartAreas>
319 <ChartLegends>
320     <ChartLegend Name="Default">
321         <Style>
322             <BackgroundGradientType>None</BackgroundGradientType>

```



```

323         <FontSize>8pt</FontSize>
324     </Style>
325     <DockToChartArea>Default</DockToChartArea>
326     <DockOutsideChartArea>true</DockOutsideChartArea>
327     <ChartLegendTitle>
328         <Caption>Categories</Caption>
329         <Style>
330             <FontSize>8pt</FontSize>
331             <FontWeight>Bold</FontWeight>
332             <TextAlign>Center</TextAlign>
333         </Style>
334     </ChartLegendTitle>
335     <HeaderSeparatorColor>Black</HeaderSeparatorColor>
336     <ColumnSeparatorColor>Black</ColumnSeparatorColor>
337 </ChartLegend>
338 </ChartLegends>
339 <ChartTitles>
340     <ChartTitle Name="Default">
341         <Caption>Sales per Product Category</Caption>
342         <Style>
343             <BackgroundGradientType>None</BackgroundGradientType>
344             <FontWeight>Bold</FontWeight>
345             <TextAlign>General</TextAlign>
346             <VerticalAlign>Top</VerticalAlign>
347         </Style>
348     </ChartTitle>
349 </ChartTitles>
350 <Palette>BrightPastel</Palette>
351 <ChartBorderSkin>
352     <Style>
353         <BackgroundColor>Gray</BackgroundColor>
354         <BackgroundGradientType>None</BackgroundGradientType>
355         <Color>White</Color>
356     </Style>
357 </ChartBorderSkin>
358 <ChartNoDataMessage Name="NoDataMessage">
359     <Caption>No Data Available</Caption>
360     <Style>
361         <BackgroundGradientType>None</BackgroundGradientType>
362         <TextAlign>General</TextAlign>
363         <VerticalAlign>Top</VerticalAlign>
364     </Style>
365 </ChartNoDataMessage>
366 <DataSetName>SalesPerProduct</DataSetName>
367 <Top>0.25708in</Top>
368 <Left>0.21542in</Left>
369 <Height>3.53125in</Height>
370 <Width>6.08333in</Width>
371 <Style>
372     <Border>
373         <Color>LightGrey</Color>
374         <Style>Solid</Style>
375     </Border>
376     <BackgroundColor>White</BackgroundColor>
377     <BackgroundGradientType>None</BackgroundGradientType>
378 </Style>
379 </Chart>
380 <Chart Name="Chart2">
381     <ChartCategoryHierarchy>
382         <ChartMembers>
383             <ChartMember>
384                 <Group Name="Chart2_CategoryGroup1">
385                     <GroupExpressions>
386                         <GroupExpression>=Fields!Year.Value</GroupExpression>
387                     </GroupExpressions>
388                 </Group>
389             </ChartMembers>
390             <ChartMember>
391                 <Group Name="Chart2_CategoryGroup2">

```

```

392         <GroupExpressions>
393             <GroupExpression>=Fields!Month.Value</GroupExpression>
394         </GroupExpressions>
395     </Group>
396     <Label>=MonthName(Fields!Month.Value)</Label>
397 </ChartMember>
398 </ChartMembers>
399     <Label>=Fields!Year.Value</Label>
400 </ChartMember>
401 </ChartMembers>
402 </ChartCategoryHierarchy>
403 <ChartSeriesHierarchy>
404     <ChartMembers>
405         <ChartMember>
406             <Label>Cost</Label>
407         </ChartMember>
408         <ChartMember>
409             <Label>Cost</Label>
410         </ChartMember>
411     </ChartMembers>
412 </ChartSeriesHierarchy>
413 <ChartData>
414     <ChartSeriesCollection>
415         <ChartSeries Name="Cost">
416             <ChartDataPoints>
417                 <ChartDataPoint>
418                     <ChartDataPointValues>
419                         <Y>=Sum(Fields!Cost.Value)</Y>
420                     </ChartDataPointValues>
421                     <ChartDataLabel>
422                         <Style />
423                     </ChartDataLabel>
424                     <Style>
425                         <Border>
426                             <Width>2pt</Width>
427                         </Border>
428                     </Style>
429                     <ChartMarker>
430                         <Style />
431                     </ChartMarker>
432                     <DataElementOutput>Output</DataElementOutput>
433                 </ChartDataPoint>
434             </ChartDataPoints>
435             <Type>Line</Type>
436             <Style />
437             <ChartEmptyPoints>
438                 <Style />
439             <ChartMarker>
440                 <Style />
441             </ChartMarker>
442             <ChartDataLabel>
443                 <Style />
444             </ChartDataLabel>
445             </ChartEmptyPoints>
446             <ValueAxisName>Primary</ValueAxisName>
447             <CategoryAxisName>Primary</CategoryAxisName>
448             <ChartSmartLabel>
449                 <CalloutLineColor>Black</CalloutLineColor>
450                 <MinMovingDistance>Opt</MinMovingDistance>
451             </ChartSmartLabel>
452         </ChartSeries>
453     <ChartSeries Name="Cost1">
454         <ChartDataPoints>
455             <ChartDataPoint>
456                 <ChartDataPointValues>
457                     <Y>=Sum(Fields!Cost.Value) / Sum(Fields!Quantity.Value)</Y>
458                 </ChartDataPointValues>
459                 <ChartDataLabel>
460                     <Style />

```

```

461         </ChartDataLabel>
462         <Style />
463         <ChartMarker>
464             <Style />
465         </ChartMarker>
466         <DataElementOutput>Output</DataElementOutput>
467     </ChartDataPoint>
468 </ChartDataPoints>
469 <Type>Line</Type>
470 <Style />
471 <ChartEmptyPoints>
472     <Style />
473     <ChartMarker>
474         <Style />
475     </ChartMarker>
476     <ChartDataLabel>
477         <Style />
478     </ChartDataLabel>
479 </ChartEmptyPoints>
480 <ChartItemInLegend>
481     <LegendText>Cost/Quantity</LegendText>
482 </ChartItemInLegend>
483 <ChartAreaName>ChartArea2</ChartAreaName>
484 <ValueAxisName>Primary</ValueAxisName>
485 <CategoryAxisName>Primary</CategoryAxisName>
486 <ChartSmartLabel>
487     <CalloutLineColor>Black</CalloutLineColor>
488     <MinMovingDistance>0pt</MinMovingDistance>
489 </ChartSmartLabel>
490 </ChartSeries>
491 </ChartSeriesCollection>
492 <ChartDerivedSeriesCollection>
493     <ChartDerivedSeries>
494         <ChartSeries Name="Series1">
495             <Type>Line</Type>
496             <Style>
497                 <Border>
498                     <Width>1.5pt</Width>
499                 </Border>
500             </Style>
501             <ChartMarker>
502                 <Style />
503             </ChartMarker>
504             <ChartDataLabel>
505                 <Style />
506             </ChartDataLabel>
507             <ChartEmptyPoints>
508                 <Style />
509                 <ChartMarker>
510                     <Style />
511                 </ChartMarker>
512                 <ChartDataLabel>
513                     <Style />
514                 </ChartDataLabel>
515             </ChartEmptyPoints>
516             <ChartItemInLegend>
517                 <LegendText>Performance (Cost)</LegendText>
518             </ChartItemInLegend>
519             <ChartAreaName>Default</ChartAreaName>
520             <ValueAxisName>Secondary</ValueAxisName>
521             <CategoryAxisName>Primary</CategoryAxisName>
522             <ChartSmartLabel>
523                 <CalloutLineColor>Black</CalloutLineColor>
524                 <MinMovingDistance>0pt</MinMovingDistance>
525             </ChartSmartLabel>
526         </ChartSeries>
527         <SourceChartSeriesName>Cost</SourceChartSeriesName>
528         <DerivedSeriesFormula>Performance</DerivedSeriesFormula>
529     </ChartDerivedSeries>

```

```

530     </ChartDerivedSeriesCollection>
531 </ChartData>
532 <ChartAreas>
533     <ChartArea Name="Default">
534         <ChartCategoryAxes>
535             <ChartAxis Name="Primary">
536                 <Style>
537                     <FontSize>8pt</FontSize>
538                 </Style>
539                 <ChartAxisTitle>
540                     <Caption />
541                     <Style>
542                         <FontSize>8pt</FontSize>
543                     </Style>
544                 </ChartAxisTitle>
545                 <Interval>3</Interval>
546                 <IntervalOffset>1</IntervalOffset>
547                 <ChartMajorGridLines>
548                     <Enabled>True</Enabled>
549                     <Style>
550                         <Border>
551                             <Color>Gainsboro</Color>
552                         </Border>
553                     </Style>
554                     <Interval>1</Interval>
555                 </ChartMajorGridLines>
556                 <ChartMinorGridLines>
557                     <Style>
558                         <Border>
559                             <Color>Gainsboro</Color>
560                             <Style>Dotted</Style>
561                         </Border>
562                     </Style>
563                 </ChartMinorGridLines>
564                 <ChartMinorTickMarks>
565                     <Length>0.5</Length>
566                 </ChartMinorTickMarks>
567                 <CrossAt>NaN</CrossAt>
568                 <Minimum>NaN</Minimum>
569                 <Maximum>NaN</Maximum>
570             </ChartAxis>
571             <ChartAxis Name="Secondary">
572                 <Style>
573                     <FontSize>8pt</FontSize>
574                 </Style>
575                 <ChartAxisTitle>
576                     <Caption>Axis Title</Caption>
577                     <Style>
578                         <FontSize>8pt</FontSize>
579                     </Style>
580                 </ChartAxisTitle>
581                 <ChartMajorGridLines>
582                     <Enabled>False</Enabled>
583                     <Style>
584                         <Border>
585                             <Color>Gainsboro</Color>
586                         </Border>
587                     </Style>
588                 </ChartMajorGridLines>
589                 <ChartMinorGridLines>
590                     <Style>
591                         <Border>
592                             <Color>Gainsboro</Color>
593                             <Style>Dotted</Style>
594                         </Border>
595                     </Style>
596                 </ChartMinorGridLines>
597                 <ChartMinorTickMarks>
598                     <Length>0.5</Length>

```

```

599         </ChartMinorTickMarks>
600         <CrossAt>NaN</CrossAt>
601         <Location>Opposite</Location>
602         <Minimum>NaN</Minimum>
603         <Maximum>NaN</Maximum>
604     </ChartAxis>
605 </ChartCategoryAxes>
606 <ChartValueAxes>
607     <ChartAxis Name="Primary">
608         <Style>
609             <FontSize>8pt</FontSize>
610         </Style>
611         <ChartAxisTitle>
612             <Caption />
613             <Style>
614                 <FontSize>8pt</FontSize>
615             </Style>
616         </ChartAxisTitle>
617         <ChartMajorGridLines>
618             <Style>
619                 <Border>
620                     <Color>Gainsboro</Color>
621                 </Border>
622             </Style>
623         </ChartMajorGridLines>
624         <ChartMinorGridLines>
625             <Style>
626                 <Border>
627                     <Color>Gainsboro</Color>
628                     <Style>Dotted</Style>
629                 </Border>
630             </Style>
631         </ChartMinorGridLines>
632         <ChartMinorTickMarks>
633             <Length>0.5</Length>
634         </ChartMinorTickMarks>
635         <CrossAt>NaN</CrossAt>
636         <Minimum>NaN</Minimum>
637         <Maximum>NaN</Maximum>
638     </ChartAxis>
639     <ChartAxis Name="Secondary">
640         <Style>
641             <FontSize>8pt</FontSize>
642         </Style>
643         <ChartAxisTitle>
644             <Caption />
645             <Style>
646                 <FontSize>8pt</FontSize>
647             </Style>
648         </ChartAxisTitle>
649         <ChartMajorGridLines>
650             <Style>
651                 <Border>
652                     <Color>Gainsboro</Color>
653                 </Border>
654             </Style>
655         </ChartMajorGridLines>
656         <ChartMinorGridLines>
657             <Style>
658                 <Border>
659                     <Color>Gainsboro</Color>
660                     <Style>Dotted</Style>
661                 </Border>
662             </Style>
663         </ChartMinorGridLines>
664         <ChartMinorTickMarks>
665             <Length>0.5</Length>
666         </ChartMinorTickMarks>
667         <CrossAt>NaN</CrossAt>

```

```

668         <Location>Opposite</Location>
669         <Minimum>NaN</Minimum>
670         <Maximum>NaN</Maximum>
671     </ChartAxis>
672 </ChartValueAxes>
673 <Style>
674     <BackgroundGradientType>None</BackgroundGradientType>
675 </Style>
676 </ChartArea>
677 <ChartArea Name="ChartArea2">
678     <ChartCategoryAxes>
679         <ChartAxis Name="Primary">
680             <Style>
681                 <FontSize>8pt</FontSize>
682             </Style>
683             <ChartAxisTitle>
684                 <Caption />
685             <Style>
686                 <FontSize>8pt</FontSize>
687             </Style>
688         </ChartAxisTitle>
689         <ChartMajorGridLines>
690             <Enabled>True</Enabled>
691             <Style>
692                 <Border>
693                     <Color>Gainsboro</Color>
694                 </Border>
695             </Style>
696             <Interval>1</Interval>
697         </ChartMajorGridLines>
698         <ChartMinorGridLines>
699             <Style>
700                 <Border>
701                     <Color>Gainsboro</Color>
702                     <Style>Dotted</Style>
703                 </Border>
704             </Style>
705         </ChartMinorGridLines>
706         <ChartMajorTickMarks>
707             <Enabled>False</Enabled>
708         </ChartMajorTickMarks>
709         <ChartMinorTickMarks>
710             <Length>0.5</Length>
711         </ChartMinorTickMarks>
712         <CrossAt>NaN</CrossAt>
713         <Minimum>NaN</Minimum>
714         <Maximum>NaN</Maximum>
715         <HideLabels>true</HideLabels>
716     </ChartAxis>
717     <ChartAxis Name="Secondary">
718         <Visible>True</Visible>
719         <Style>
720             <FontSize>8pt</FontSize>
721         </Style>
722         <ChartAxisTitle>
723             <Caption />
724         <Style>
725             <FontSize>8pt</FontSize>
726         </Style>
727     </ChartAxisTitle>
728     <ChartMajorGridLines>
729         <Enabled>False</Enabled>
730     <Style>
731         <Border>
732             <Color>Gainsboro</Color>
733         </Border>
734     </Style>
735 </ChartMajorGridLines>
736 <ChartMinorGridLines>

```

```

737         <Style>
738             <Border>
739                 <Color>Gainsboro</Color>
740                 <Style>Dotted</Style>
741             </Border>
742         </Style>
743     </ChartMinorGridLines>
744     <ChartMajorTickMarks>
745         <Enabled>True</Enabled>
746         <Interval>12</Interval>
747         <IntervalOffset>=First (Fields!Month.Value)</IntervalOffset>
748     </ChartMajorTickMarks>
749     <ChartMinorTickMarks>
750         <Length>0.5</Length>
751     </ChartMinorTickMarks>
752     <CrossAt>NaN</CrossAt>
753     <Location>Opposite</Location>
754     <Minimum>NaN</Minimum>
755     <Maximum>NaN</Maximum>
756     <HideLabels>>true</HideLabels>
757 </ChartAxis>
758 </ChartCategoryAxes>
759 <ChartValueAxes>
760     <ChartAxis Name="Primary">
761         <Style>
762             <FontSize>8pt</FontSize>
763             <Format>#,0.00; (#,0.00)</Format>
764         </Style>
765         <ChartAxisTitle>
766             <Caption />
767         <Style>
768             <FontSize>8pt</FontSize>
769         </Style>
770     </ChartAxisTitle>
771     <ChartMajorGridLines>
772         <Style>
773             <Border>
774                 <Color>Gainsboro</Color>
775             </Border>
776         </Style>
777     </ChartMajorGridLines>
778     <ChartMinorGridLines>
779         <Style>
780             <Border>
781                 <Color>Gainsboro</Color>
782                 <Style>Dotted</Style>
783             </Border>
784         </Style>
785     </ChartMinorGridLines>
786     <ChartMinorTickMarks>
787         <Length>0.5</Length>
788     </ChartMinorTickMarks>
789     <CrossAt>NaN</CrossAt>
790     <Minimum>NaN</Minimum>
791     <Maximum>NaN</Maximum>
792 </ChartAxis>
793 <ChartAxis Name="Secondary">
794     <Style>
795         <FontSize>8pt</FontSize>
796     </Style>
797     <ChartAxisTitle>
798         <Caption />
799     <Style>
800         <FontSize>8pt</FontSize>
801     </Style>
802 </ChartAxisTitle>
803 <ChartMajorGridLines>
804     <Style>
805         <Border>

```

```

806         <Color>Gainsboro</Color>
807     </Border>
808 </Style>
809 </ChartMajorGridLines>
810 <ChartMinorGridLines>
811     <Style>
812         <Border>
813             <Color>Gainsboro</Color>
814             <Style>Dotted</Style>
815         </Border>
816     </Style>
817 </ChartMinorGridLines>
818 <ChartMinorTickMarks>
819     <Length>0.5</Length>
820 </ChartMinorTickMarks>
821 <CrossAt>NaN</CrossAt>
822 <Location>Opposite</Location>
823 <Minimum>NaN</Minimum>
824 <Maximum>NaN</Maximum>
825 </ChartAxis>
826 </ChartValueAxes>
827 <Style>
828     <BackgroundGradientType>None</BackgroundGradientType>
829 </Style>
830 <AlignOrientation>Vertical</AlignOrientation>
831 <ChartAlignType>
832     <InnerPlotPosition>true</InnerPlotPosition>
833 </ChartAlignType>
834 <AlignWithChartArea>Default</AlignWithChartArea>
835 </ChartArea>
836 </ChartAreas>
837 <ChartLegends>
838     <ChartLegend Name="Default">
839         <Style>
840             <BackgroundGradientType>None</BackgroundGradientType>
841             <FontSize>8pt</FontSize>
842         </Style>
843         <Position>TopLeft</Position>
844         <DockOutsideChartArea>true</DockOutsideChartArea>
845         <ChartLegendTitle>
846             <Caption />
847             <Style>
848                 <FontSize>8pt</FontSize>
849                 <FontWeight>Bold</FontWeight>
850                 <TextAlign>Center</TextAlign>
851             </Style>
852         </ChartLegendTitle>
853         <HeaderSeparatorColor>Black</HeaderSeparatorColor>
854         <ColumnSeparatorColor>Black</ColumnSeparatorColor>
855     </ChartLegend>
856 </ChartLegends>
857 <ChartTitles>
858     <ChartTitle Name="Default">
859         <Caption>Costs per Month</Caption>
860         <Style>
861             <BackgroundGradientType>None</BackgroundGradientType>
862             <FontWeight>Bold</FontWeight>
863             <TextAlign>General</TextAlign>
864             <VerticalAlign>Top</VerticalAlign>
865         </Style>
866     </ChartTitle>
867 </ChartTitles>
868 <Palette>BrightPastel</Palette>
869 <ChartBorderSkin>
870     <Style>
871         <BackgroundColor>Gray</BackgroundColor>
872         <BackgroundGradientType>None</BackgroundGradientType>
873         <Color>White</Color>
874     </Style>

```



```

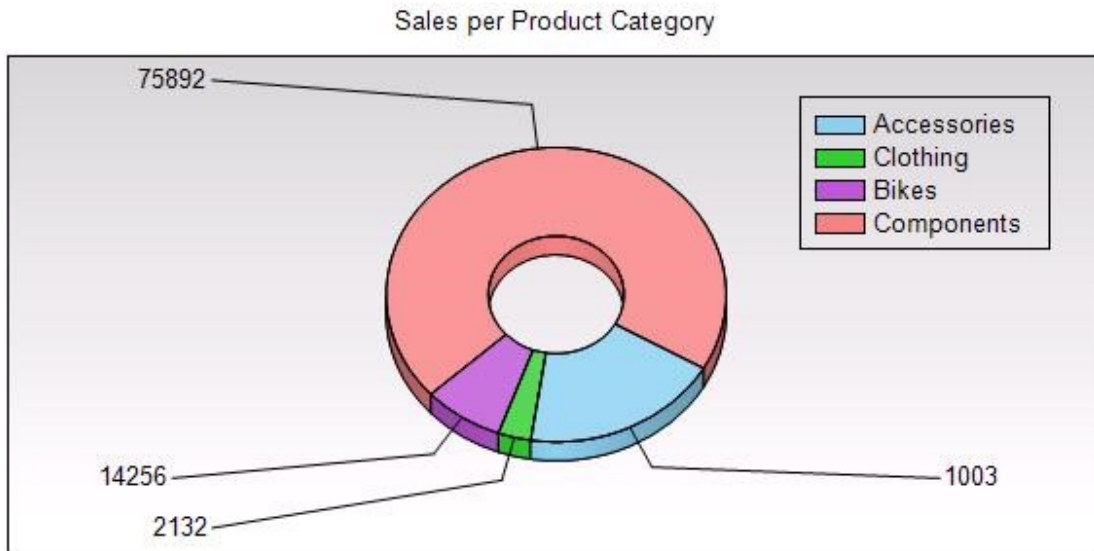
875     </ChartBorderSkin>
876     <ChartNoDataMessage Name="NoDataMessage">
877         <Caption>No Data Available</Caption>
878         <Style>
879             <BackgroundGradientType>None</BackgroundGradientType>
880             <TextAlign>General</TextAlign>
881             <VerticalAlign>Top</VerticalAlign>
882         </Style>
883     </ChartNoDataMessage>
884     <DataSetName>CostsPerMonth</DataSetName>
885     <Top>3.99667in</Top>
886     <Left>0.21542in</Left>
887     <Height>4.60417in</Height>
888     <Width>6.08333in</Width>
889     <ZIndex>1</ZIndex>
890     <Style>
891         <Border>
892             <Color>LightGrey</Color>
893             <Style>Solid</Style>
894         </Border>
895         <BackgroundColor>White</BackgroundColor>
896         <BackgroundGradientType>None</BackgroundGradientType>
897     </Style>
898 </Chart>
899 </ReportItems>
900 <Height>8.80917in</Height>
901 <Style />
902 </Body>
903 <Width>6.5175in</Width>
904 <Page>
905     <LeftMargin>1in</LeftMargin>
906     <RightMargin>1in</RightMargin>
907     <TopMargin>1in</TopMargin>
908     <BottomMargin>1in</BottomMargin>
909 <Style />
910 </Page>
911 <rd:ReportID>19e8862d-b091-4af2-affb-3de14da9fcbe</rd:ReportID>
912 <rd:ReportUnitType>Inch</rd:ReportUnitType>
913</Report>

```

### 3.5 Doughnut Chart

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The following sample report shows a doughnut **chart**.



**Figure 14: Doughnut chart example**

The following table shows the data that this chart is bound to.

Product category name	Year	Quantity
Accessories	2001	1003
	2002	27207
	2003	23734
	2003	103978
	2004	292660
Bikes	2001	14256
	2002	49810
	2003	30773
	2003	43241
	2004	42456
Clothing	2001	2132
	2002	16927
	2003	16515
	2003	18862
	2004	19234
Components	2001	75892
	2002	332885
	2003	266860

Product category name	Year	Quantity
	2003	314930
	2004	661065

This chart shows the sum of the quantity for each product category regardless of the year. The following RDL snippet shows how the chart is specified to display data the way it looks in the preceding picture.

```
<Chart Name="chart1">
  <DataSetName>SalesPerProduct</DataSetName>
  <PointWidth>0</PointWidth>
  <Palette>Pastel</Palette>
  <Subtype>Plain</Subtype>
  <Type>Doughnut</Type>
  <Top>0.125in</Top>
  <Left>0.125in</Left>
  <Width>6.25in</Width>
  <Height>3.125in</Height>
  <CategoryAxis />
  <ValueAxis />
```

This is a doughnut chart bound to a [DataSet](#) that is called "SalesPerProduct" with a Pastel palette.

```
<SeriesGroupings>
  <SeriesGrouping>
    <DynamicSeries>
      <Grouping Name="chart1 SeriesGroup1">
        <GroupExpressions>
          <GroupExpression>=Fields!ProductCategoryName.Value</GroupExpression>
        </GroupExpressions>
      </Grouping>
      <Label>=Fields!ProductCategoryName.Value</Label>
    </DynamicSeries>
  </SeriesGrouping>
</SeriesGroupings>
```

The series grouping specifies that each Product Category value is in the legend. Note that the [Label](#) element specifies the text of the item in the legend. The number of the series groups is the number of slices in the chart. The sizes of the slices are determined by the following value.

```
<ChartData>
  <ChartSeries>
    <DataPoints>
      <DataPoint>
        <DataValues>
          <DataValue>
            <Value>=Sum(Fields!Quantity.Value)</Value>
          </DataValue>
        </DataValues>
        <DataLabel>
          <Style />
          <Value>=Fields!Quantity.Value</Value>
          <Position>BottomRight</Position>
          <Visible>true</Visible>
        </DataLabel>
      </DataPoint>
    </DataPoints>
  </ChartSeries>
</ChartData>
```

As shown in the figure, each slice has its value displayed as a [DataLabel](#) element as specified in the above RDL.

```
<Legend>
  <Visible>true</Visible>
  <Style>
    <BorderStyle>
      <Default>Solid</Default>
    </BorderStyle>
  </Style>
  <InsidePlotArea>true</InsidePlotArea>
</Legend>
```

The [Legend](#) element is displayed inside the [PlotArea](#) element with the gradient background color specified by the **PlotArea** element in the following RDL.

```
<PlotArea>
  <Style>
    <BackgroundColor>LightGrey</BackgroundColor>
    <BackgroundGradientType>TopBottom</BackgroundGradientType>
    <BorderStyle>
      <Default>Solid</Default>
    </BorderStyle>
  </Style>
</PlotArea>
```

Because this is a 3D chart, the [ThreeDProperties](#) element is required as well in the following RDL.

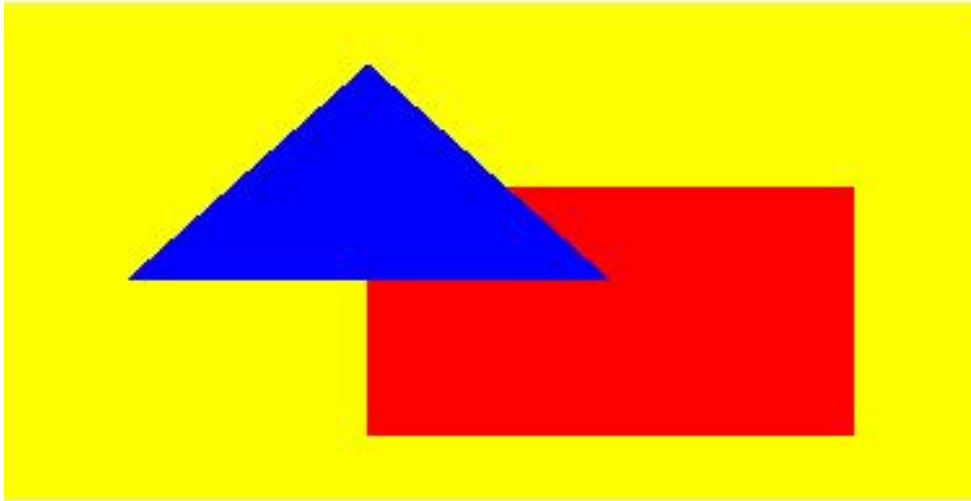
```
<ThreeDProperties>
  <Enabled>true</Enabled>
  <Rotation>30</Rotation>
  <Inclination>30</Inclination>
  <Shading>Real</Shading>
  <WallThickness>50</WallThickness>
</ThreeDProperties>
```

The following RDL specifies the chart title text and the chart style.

```
<Style>
  <BackgroundColor>White</BackgroundColor>
</Style>
<Title>
  <Caption>Sales per Product Category</Caption>
</Title>
</Chart>
```

### 3.6 CustomReportItem

The following is an example RDL definition of a [CustomReportItem](#) for Schema version [RDL 2008/01](#) of the element, with its data. The example is based on the Microsoft Polygons example available at [\[MSDN-RDLDT\]](#). The rendered snippet is pictured in the following figure.



**Figure 15: A data source is defined as embedded XML.**

```
<DataSources>
  <DataSource Name="PolygonsData">
    <ConnectionProperties>
      <DataProvider>XML</DataProvider>
      <ConnectionString />
    </ConnectionProperties>
  </DataSource>
</DataSources>
```

A single [DataSet](#) element is defined with five [Fields](#) elements, named in both the [Query](#) and the [Report](#) as "Object", "Color", "Point", "X", and "Y".

```
<DataSets>
  <DataSet Name="DataSet1">
    <Fields>
      <Field Name="Object">
        <DataField>Object</DataField>
      </Field>
      <Field Name="Color">
        <DataField>Color</DataField>
      </Field>
      <Field Name="Point">
        <DataField>Point</DataField>
      </Field>
      <Field Name="X">
        <DataField>X</DataField>
      </Field>
      <Field Name="Y">
        <DataField>Y</DataField>
      </Field>
    </Fields>
    <Query>
      <DataSourceName>PolygonsData</DataSourceName>
```

The [Query.CommandText](#) element contains the XML. Note that the XML brackets ("[<](#)" and "[>](#)") have had to be replaced with encoded versions ("[&lt;](#)" and "[&gt;](#)") to avoid confusing an RDL reader. The following table illustrates the data.

Object	Color	Point	X	Y
1	Red	1	75	75
1	Red	2	75	175
1	Red	3	175	175
1	Red	4	175	75
2	Blue	1	75	25
2	Blue	2	25	112
2	Blue	3	125	112

**Figure 16: Query.CommandText data table example**

```

<CommandText>
  <Query>
    <XmlData>
      <Rows>
        <Row>
          <Object>1</Object> <Color>Red</Color>
          <Point>1</Point> <X>75</X> <Y>75</Y>
        </Row>
        <Row>
          <Object>1</Object> <Color>Red</Color>
          <Point>2</Point> <X>75</X> <Y>175</Y>
        </Row>
        <Row>
          <Object>1</Object> <Color>Red</Color>
          <Point>3</Point> <X>175</X> <Y>175</Y>
        </Row>
        <Row>
          <Object>1</Object> <Color>Red</Color>
          <Point>4</Point> <X>175</X> <Y>75</Y>
        </Row>
        <Row>
          <Object>2</Object> <Color>Blue</Color>
          <Point>1</Point> <X>75</X> <Y>25</Y>
        </Row>
        <Row>
          <Object>2</Object> <Color>Blue</Color>
          <Point>2</Point> <X>25</X> <Y>112</Y>
        </Row>
        <Row>
          <Object>2</Object> <Color>Blue</Color>
          <Point>3</Point> <X>125</X> <Y>112</Y>
        </Row>
      </Rows>
    </XmlData>
    <ElementPath>Rows/Row{Object (Integer), Color (String), Point (Integer), X (Integer), Y (Integer)}</ElementPath>
  </Query>
</CommandText>
</Query>
</DataSet>
</DataSets>

```

The **CustomReportItem** element has its [CustomReportItem.Name](#) attribute set to "customReportItem1". It is set to 0.125 inches from the top of its containing object (either the [Body](#) or a report item) and, because no [CustomReportItem.Left](#) element is defined, the position of its left is the position of the left of its containing object. The custom report item will be 2 inches high and 10 centimeters wide.

```
<CustomReportItem Name="customReportItem1">
```

```
<Top>0.125in</Top>
<Height>2in</Height>
<Width>10cm</Width>
```

The [CustomReportItem.Type](#) of the element is set to a name recognizable by the rendering engine, so that custom code to render this item can be utilized. The [CustomData.DataSetName](#) element points to the prepared data.

```
<Type>Polygons</Type>
<CustomData>
  <DataSetName>DataSet1</DataSetName>
```

A [CustomData.DataColumnHierarchy](#) element is required to be defined, although this example only has one column hierarchy, which covers all rows.

```
<DataColumnHierarchy>
  <DataMembers>
    <DataMember />
  </DataMembers>
</DataColumnHierarchy>
```

The [DataRowHierarchy](#) element is also mandatory and in this case uses a two-level hierarchy to separate the two objects, so that the lower hierarchy is processed for each object in turn.

```
<DataRowHierarchy>
  <DataMembers>
```

The outer [DataMember](#) element is specified with a [Group](#) that is named "Object", which groups on the value of the Object field. The **DataMember** has two [CustomProperty](#) elements defined. The name of these custom properties is only relevant to the custom code, which looks for and matches "poly:Color" and "poly:Hyperlink". The value of the first **custom property** is set to the value of the Color field from the data set; the value of the second custom property is set to a URL that is intended to be set as the [Action](#) of an [Image](#) that the custom code will create.

```
<DataMember>
  <Group Name="Object">
    <GroupExpressions>
      <GroupExpression>=Fields!Object.Value</GroupExpression>
    </GroupExpressions>
  </Group>
  <CustomProperties>
    <CustomProperty>
      <Name>poly:Color</Name>
      <Value>=Fields!Color.Value</Value>
    </CustomProperty>
    <CustomProperty>
      <Name>poly:Hyperlink</Name>
      <Value>http://microsoft.com</Value>
    </CustomProperty>
  </CustomProperties>
```

Within the outer DataMember element, an inner [DataMembers](#) collection is specified. A single **DataMember** descendant is specified.

```
<DataMembers>
```

```
<DataMember>
```

A Group element is specified so that a sort can be defined. Since the data is known to be good, no grouping expression is specified. The data is sorted on the value of the Point field.

```
    <Group Name="Point" />
    <SortExpressions>
      <SortExpression>
        <Value>=Fields!Point.Value</Value>
      </SortExpression>
    </SortExpressions>
  </DataMember>
</DataMembers>
</DataMember>
</DataMembers>
</DataRowHierarchy>
```

There is only one leaf **DataMember** in the **DataRowHierarchy**, so there is only one [DataRow](#) element specified. The row has two cells in it, specifying X and Y data based on the value of the X and Y fields.

```
<DataRows>
  <DataRow>
    <DataCell>
      <DataValue>
        <Name>X</Name>
        <Value>=Fields!X.Value</Value>
      </DataValue>
      <DataValue>
        <Name>Y</Name>
        <Value>=Fields!Y.Value</Value>
      </DataValue>
    </DataCell>
  </DataRow>
</DataRows>
</CustomData>
```

Further values are specified for the custom code by using a [CustomProperties](#) element to define four name/value pairs. Again, the value of the [Name](#) element is required to be matched in code and has no relevance to any namespace. The value of each [Value](#) element could be expressed as a literal or as an expression.

```
<CustomProperties>
  <CustomProperty>
    <Name>poly:MaxX</Name>
    <Value>200</Value>
  </CustomProperty>
  <CustomProperty>
    <Name>poly:MinX</Name>
    <Value>0</Value>
  </CustomProperty>
  <CustomProperty>
    <Name>poly:MaxY</Name>
    <Value>200</Value>
  </CustomProperty>
  <CustomProperty>
    <Name>poly:MinY</Name>
    <Value>0</Value>
  </CustomProperty>
</CustomProperties>
```



A [Style](#) element specifies a background color that the custom code can use. It also specifies a default foreground color for the custom code to use. The empty [Border](#) element has no effect.

```
<Style>
  <Border />
  <BackgroundColor>Yellow</BackgroundColor>
  <Color>Red</Color>
</Style>

</CustomReportItem>
```

### 3.7 Tablix 1

The following is an example RDL definition of a [Tablix](#) element that is used in a very simple manner. The **tablix** acts as a container for a [Rectangle](#); the **rectangle** contains the **report layout**. This layout provides similar functionality to the [List](#) element in schema versions prior to [RDL 2008/01](#). The data is fetched from the Microsoft Adventure Works example database. An extract of the rendered XML is as follows.

#500-75 O'Connor Street Ottawa	Ontario Canada K4B 1S2
#9900 2700 Production Way Burnaby	British Columbia Canada V5A 4X1
1050 Oak Street Seattle	Washington United States 98104
1200 First Ave. Joliet	Illinois United States 60433
123 Camelia Avenue Oxnard	California United States 93030
123 W. Lake Ave. Peoria	Illinois United States 61606

**Figure 17: Tablix example 1**

The **Tablix** element has its [Tablix.Name](#) attribute set to "Tablix1". It has a single uncomplicated column, needing only the minimal [TablixColumnHierarchy](#). The [TablixRowHierarchy](#) is almost as simple, but because the single tablix row is to be repeated for each row of data, the [TablixMember](#) is defined with a [Group](#). There is no grouping, filtering, or sorting to be done on the **Group**, which defines it as a **detail group**.

```
<Tablix Name="Tablix1">
  <TablixColumnHierarchy>
    <TablixMembers>
      <TablixMember />
    </TablixMembers>
```

```

</TablixColumnHierarchy>
<TablixRowHierarchy>
  <TablixMembers>
    <TablixMember>
      <Group Name="Details" />
    </TablixMember>
  </TablixMembers>
</TablixRowHierarchy>

```

The single column is set to be 5.5 inches wide; therefore, the tablix is going to be 5.5 inches wide.

```

<TablixBody>
  <TablixColumns>
    <TablixColumn>
      <Width>5.5in</Width>
    </TablixColumn>
  </TablixColumns>

```

The single row specifies a single [TablixCell](#)—because there is only a single column—and this has a **Rectangle** in it. The rectangle has various controls in it. For each row of data, the cell is repeated vertically, each being 0.42 inches high. The rendered height of the tablix depends on how many rows of data are in the [DataSet](#).

```

<TablixRows>
  <TablixRow>
    <Height>0.42in</Height>
    <TablixCells>
      <TablixCell>
        <CellContents>
          <Rectangle Name="Rectangle1">
            <ReportItems>
              <Textbox Name="Address">
                <CanGrow>true</CanGrow>
                <KeepTogether>true</KeepTogether>
                <Paragraphs>
                  <Paragraph>
                    <TextRuns>
                      <TextRun>

```

The value in the text box is a data row value, so it can change for each rendered row of the tablix.

```

              <Value>=Fields!AddressLine1.Value</Value>
            </TextRun>
          </TextRuns>
        </Paragraph>
      <Paragraph>
        <TextRuns>
          <TextRun>
            <Value>=Fields!City.Value</Value>
          </TextRun>
        </TextRuns>
      </Paragraph>
    </Paragraphs>
  <Height>0.23958in</Height>
  <Width>2.5625in</Width>
</Textbox>
<Textbox Name="StateProvince">
  <CanGrow>true</CanGrow>
  <KeepTogether>true</KeepTogether>
  <Paragraphs>
    <Paragraph>
      <TextRuns>

```

```

        <TextRun>
            <Value>=Fields!StateProvince.Value</Value>
        </TextRun>
    </TextRuns>
</Paragraph>
<Paragraph>
    <TextRuns>
        <TextRun>
            <Value>=Fields!CountryRegion.Value</Value>
        </TextRun>
    </TextRuns>
</Paragraph>
<Paragraph>
    <TextRuns>
        <TextRun>
            <Value>=Fields!PostalCode.Value</Value>
        </TextRun>
    </TextRuns>
</Paragraph>
</Paragraphs>
<Left>3in</Left>
<Height>0.22in</Height>
<Width>2.6in</Width>
<ZIndex>1</ZIndex>
</Textbox>
<Line Name="Line1">
    <Top>0.35in</Top>
    <Left>0.15in</Left>
    <Height>0in</Height>
    <Width>5.2in</Width>
    <ZIndex>2</ZIndex>
    <Style>
        <Border>
            <Style>Solid</Style>
        </Border>
    </Style>
</Line>
</ReportItems>
<KeepTogether>true</KeepTogether>
</Rectangle>
</CellContents>
</TablixCell>
</TablixCells>
</TablixRow>
</TablixRows>
</TablixBody>

```

If there is more than one **DataSet** in the report, the [Tablix.DataSetName](#) element is required to be defined; otherwise it would default to the single **DataSet** available.

```

    <DataSetName>AddressDataSet</DataSetName>
</Tablix>

```

### 3.8 Tablix 2

The following is an example RDL definition of a [Tablix](#) element that is used to lay out data in a straightforward manner. The tablix uses text boxes within cells to format and display the data. This layout provides similar functionality to the [Table](#) element in versions of RDL prior to 2008/01. The data is fetched from the Microsoft Adventure Works example database. The top of the rendered definition is shown in the following figure.

Address Line1	City	Country
New Millhouse, 2583 Milton Park	<b>Abingdon</b> England OX14 4SE	United Kingdom
2255 254th Avenue Se	<b>Albany</b> Oregon 97321	United States
Heritage Mall	<b>Albany</b> Oregon 97321	United States
Vista Marketplace	<b>Alhambra</b> California 91801	United States
Viejas Outlet Center	<b>Alpine</b> California 91901	United States
Lincoln Square	<b>Arlington</b> Texas 76010	United States
99700 Bell Road	<b>Auburn</b> California 95603	United States
32605 West 252 Mile Road, Suite 250	<b>Aurora</b> Ontario L4G 7N6	Canada

**Figure 18: Tablix example 2**

The **Tablix** element has its [Tablix.Name](#) attribute set to "Tablix2". It is positioned 1 inch from the left edge of its container.

```
<Tablix Name="Tablix2">
  <Left>1in</Left>
```

The entire tablix is rendered only if the value of the *ShowTable* parameter is set to false. If it is rendered, a page break is wanted before the tablix appears, and after the last row of the rendered tablix, another page break is wanted.

```
<Visibility>
  <Hidden>=Parameters!ShowTable.Value=False</Hidden>
</Visibility>
<PageBreak>
  <BreakLocation>StartAndEnd</BreakLocation>
</PageBreak>
```

There are three **static columns** in this tablix, so three [TablixCell](#) elements are required in each row (or fewer if one of the cells spans more than one column).

```
<TablixColumnHierarchy>
  <TablixMembers>
    <TablixMember />
    <TablixMember />
    <TablixMember />
  </TablixMembers>
</TablixColumnHierarchy>
```

The [TablixRowHierarchy](#) element specifies two rows. The first is a **static row** that is to be kept with the **dynamic row** that follows. This row is to behave as a **column header**. Therefore, the row has a [TablixMember.KeepWithGroup](#) element that has a value of After and a [TablixMember.RepeatOnNewPage](#) element that has a value of true.

```
<TablixRowHierarchy>
  <TablixMembers>
    <TablixMember>
      <KeepWithGroup>After</KeepWithGroup>
      <RepeatOnNewPage>>true</RepeatOnNewPage>
    </TablixMember>
  </TablixMembers>
</TablixRowHierarchy>
```

The second row is a **detail group** with a [SortExpression](#) that causes the data in the group to appear in order of city name, A–Z.

```
<TablixMember>
  <Group Name="Details" />
  <SortExpressions>
    <SortExpression>
      <Value>=Fields!City.Value</Value>
    </SortExpression>
  </SortExpressions>
</TablixMember>
</TablixMembers>
</TablixRowHierarchy>
```

The three columns defined by the [TablixColumnHierarchy](#) are set up with specific widths.

```
<TablixBody>
  <TablixColumns>
    <TablixColumn>
      <Width>2in</Width>
    </TablixColumn>
    <TablixColumn>
      <Width>1.5in</Width>
    </TablixColumn>
    <TablixColumn>
      <Width>3cm</Width>
    </TablixColumn>
  </TablixColumns>
```

The first row is set to 0.3 inches high and contains three cells. Each cell has an aqua background and has a light gray border, created by setting the properties on the descendant text boxes. The text in each text box is set as a header to the columns, Address, City, and Country.

```
<TablixRows>
  <TablixRow>
```

```

<Height>0.3in</Height>
<TablixCells>
  <TablixCell>
    <CellContents>
      <Textbox Name="Heading1">
        <KeepTogether>true</KeepTogether>
        <Paragraphs>
          <Paragraph>
            <TextRuns>
              <TextRun>
                <Value>Address</Value>
              </TextRun>
            </TextRuns>
          </Paragraph>
        </Paragraphs>
      </Textbox>
    </CellContents>
  </TablixCell>
  <TablixCell>
    <CellContents>
      <Textbox Name="Heading2">
        <Paragraphs>
          <Paragraph>
            <TextRuns>
              <TextRun>
                <Value>City</Value>
              </TextRun>
            </TextRuns>
          </Paragraph>
        </Paragraphs>
      </Textbox>
    </CellContents>
  </TablixCell>
  <TablixCell>
    <CellContents>
      <Textbox Name="Heading3">
        <KeepTogether>true</KeepTogether>
        <Paragraphs>
          <Paragraph>
            <TextRuns>
              <TextRun>
                <Value>Country</Value>
              </TextRun>
            </TextRuns>
          </Paragraph>
        </Paragraphs>
      </Textbox>
    </CellContents>
  </TablixCell>

```

```

    </TablixCell>
  </TablixCells>
</TablixRow>

```

The second row is associated with the row hierarchy that is a detail group; it will therefore be repeated for each row of data. The row is set to 0.6 inches high, but this will increase if the data needs more space because the descendant [Textbox](#) elements have the [Textbox.CanGrow](#) element specified with a value of true. As with the headings, each cell will appear to have a solid, light gray border because each descendant text box has that border defined.

```

<TablixRow>
  <Height>0.6in</Height>
  <TablixCells>
    <TablixCell>
      <CellContents>
        <Textbox Name="Address">
          <CanGrow>true</CanGrow>
          <KeepTogether>true</KeepTogether>
          <Paragraphs>
            <Paragraph>
              <TextRuns>
                <TextRun>
                  <Value>=Fields!AddressLine1.Value</Value>
                </TextRun>
              </TextRuns>
            </Paragraph>
            <Paragraph>
              <TextRuns>
                <TextRun>
                  <Value>=Fields!AddressLine2.Value</Value>
                </TextRun>
              </TextRuns>
            </Paragraph>
          </Paragraphs>
          <Style>
            <Border>
              <Color>LightGrey</Color>
              <Style>Solid</Style>
            </Border>
            <PaddingLeft>2pt</PaddingLeft>
          </Style>
        </Textbox>
      </CellContents>
    </TablixCell>
    <TablixCell>
      <CellContents>
        <Textbox Name="Area">
          <CanGrow>true</CanGrow>
          <KeepTogether>true</KeepTogether>
          <Paragraphs>
            <Paragraph>
              <TextRuns>
                <TextRun>
                  <Value>=Fields!City.Value</Value>
                  <Style>
                    <FontWeight>Bold</FontWeight>
                  </Style>
                </TextRun>
              </TextRuns>
            </Paragraph>
            <Paragraph>
              <TextRuns>
                <TextRun>
                  <Value>=Fields!StateProvince.Value</Value>
                </TextRun>
              </TextRuns>
            </Paragraph>
          </Paragraphs>
        </Textbox>
      </CellContents>
    </TablixCell>
  </TablixCells>
</TablixRow>

```

```

        </Paragraph>
        <Paragraph>
            <TextRuns>
                <TextRun>
                    <Value>=Fields!PostalCode.Value</Value>
                </TextRun>
            </TextRuns>
        </Paragraph>
    </Paragraphs>
    <Style>
        <Border>
            <Color>LightGrey</Color>
            <Style>Solid</Style>
        </Border>
        <PaddingLeft>2pt</PaddingLeft>
    </Style>
</Textbox>
</CellContents>
</TablixCell>
<TablixCell>
    <CellContents>
        <Textbox Name="Country">
            <CanGrow>true</CanGrow>
            <KeepTogether>true</KeepTogether>
            <Paragraphs>
                <Paragraph>
                    <TextRuns>
                        <TextRun>
                            <Value>=Fields!CountryRegion.Value</Value>
                        </TextRun>
                    </TextRuns>
                </Paragraph>
            </Paragraphs>
            <Style>
                <Border>
                    <Color>LightGrey</Color>
                    <Style>Solid</Style>
                </Border>
                <PaddingLeft>2pt</PaddingLeft>
            </Style>
        </Textbox>
    </CellContents>
</TablixCell>
</TablixCells>
</TablixRow>
</TablixRows>
</TablixBody>

```

If possible, the entire tablix is to be rendered on a single page.

```
<KeepTogether>true</KeepTogether>
```

The entire tablix has a tooltip of "Addresses" and a solid border of the default color around it.

```

<ToolTip>Addresses</ToolTip>
<Style>
    <Border>
        <Style>Solid</Style>
    </Border>
</Style>
<DataSetName>AddressDataSet</DataSetName>
</Tablix>

```



### 3.9 Tablix 3

The following is an example RDL definition of a [Tablix](#) element used to lay out data in nested rows, repeating columns, and a sibling column. The tablix uses text boxes within cells to format and display the data. This layout encompasses the functionality of the [Matrix](#) element in schema versions prior to [RDL 2008/01](#). The data is fetched from the Microsoft Adventure Works example database. The top of the rendered definition is shown in the following illustration.

Bicycle sales for June 2004		linda		jae		Total Orders
Category	Product Name	List Price	Standard Cost	List Price	Standard Cost	
Handlebars	<b>First order received</b>	Jun 01, 2004		Jun 01, 2004		
	ML Mountain Handlebars			185.7600	82.4775	9
	HL Mountain Handlebars			240.5400	106.7998	2
	HL Road Handlebars	120.2700	53.3999	240.5400	106.7998	4
	HL Touring Handlebars	91.5700	40.6571			1
	LL Mountain Handlebars			89.0800	39.5516	5
	<b>Total</b>		211.8400	94.0570	755.9200	335.6287
Helmets	<b>First order received</b>	Jun 01, 2004		Jun 01, 2004		
	Sport-100 Helmet, Black	69.9800	26.1726	104.9700	39.2589	47
	Sport-100 Helmet, Blue	104.9700	39.2589	104.9700	39.2589	31
	Sport-100 Helmet, Red	69.9800	26.1726	139.9600	52.3452	33
<b>Total</b>		244.9300	91.6041	349.9000	130.8630	111
Hydration Packs	<b>First order received</b>	Jun 01, 2004		Jun 01, 2004		
	Hydration Pack - 70 oz.	109.9800	41.1326	219.9600	82.2652	45
	<b>Total</b>		109.9800	41.1326	219.9600	82.2652
Jerseys	<b>First order received</b>	Jun 01, 2004		Jun 01, 2004		
	Long-Sleeve Logo Jersey, M	99.9800	76.9846	149.9700	115.4769	21
	Short-Sleeve Classic Jersey, XL	107.9800	83.1446	215.9600	166.2892	42
	Long-Sleeve Logo Jersey, L	149.9700	115.4769	149.9700	115.4769	42
	Short-Sleeve Classic Jersey, S	107.9800	83.1446	161.9700	124.7169	27
	Short-Sleeve Classic Jersey, L	161.9700	124.7169	161.9700	124.7169	43
<b>Total</b>		49.9900	38.4923	99.9800	76.9846	14

Figure 19: Tablix example 3

The **Tablix** element has its [Tablix.Name](#) attribute set to "Tablix3".

```
s<Tablix Name="Tablix3">
```

The column hierarchy is defining two **dynamic columns**. The first is colored aqua and the second is colored pink.

```
<TablixColumnHierarchy>
```

The first dynamic column is grouping on SalesPerson and contains two **static columns** within the same hierarchy. All text boxes within the first tablix member are colored aqua. The dynamic column and the static columns within it are repeated for each SalesPerson in the data.

```
<TablixMembers>
  <TablixMember>
    <Group Name="SalesPerson">
      <GroupExpressions>
        <GroupExpression>=Fields!SalesPerson.Value</GroupExpression>
      </GroupExpressions>
    </Group>
  </TablixMember>
</TablixMembers>
```

Because the [TablixHeader](#) element is within the column hierarchy, the [RdlSize](#) value represents row height.

```
<Size>0.48438in</Size>
<CellContents>
  <Textbox Name="SalesPerson">
    <CanGrow>true</CanGrow>
    <KeepTogether>true</KeepTogether>
    <Paragraphs>
      <Paragraph>
        <TextRuns>
          <TextRun>
            <Value>=Fields!SalesPerson.Value</Value>
            <Style>
              <FontSize>12pt</FontSize>
            </Style>
          </TextRun>
        </TextRuns>
        <Style>
          <TextAlign>Center</TextAlign>
        </Style>
      </Paragraph>
    </Paragraphs>
    <Style>
      <Border>
        <Color>LightGrey</Color>
        <Style>Solid</Style>
      </Border>
      <BackgroundColor>Aqua</BackgroundColor>
    </Style>
  </Textbox>
</CellContents>
</TablixHeader>
```

The first nested [TablixMember](#) defines a nested column for List Price under each SalesPerson.

```
<TablixMembers>
  <TablixMember>
    <TablixHeader>
      <Size>0.45833in</Size>
      <CellContents>
        <Textbox Name="Textbox8">
          <CanGrow>true</CanGrow>
          <KeepTogether>true</KeepTogether>
          <Paragraphs>
            <Paragraph>
              <TextRuns>
                <TextRun>
                  <Value>List Price</Value>
                  <Style>
                    <FontWeight>Bold</FontWeight>
                  </Style>
                </TextRun>
              </TextRuns>
            </Paragraph>
          </Paragraphs>
        </Textbox>
      </CellContents>
    </TablixHeader>
  </TablixMember>
</TablixMembers>
```

```

        </Style>
      </TextRun>
    </TextRuns>
  <Style>
    <TextAlign>Right</TextAlign>
  </Style>
</Paragraph>
</Paragraphs>
</Textbox>
</CellContents>
</TablixHeader>
</TablixMember>

```

The second nested tablix member defines a nested column for Standard Cost under each SalesPerson.

```

<TablixMember>
  <TablixHeader>
    <Size>0.45833in</Size>
    <CellContents>
      <Textbox Name="Textbox9">
        <CanGrow>true</CanGrow>
        <KeepTogether>true</KeepTogether>
        <Paragraphs>
          <Paragraph>
            <TextRuns>
              <TextRun>
                <Value>Standard Cost</Value>
                <Style>
                  <FontWeight>Bold</FontWeight>
                </Style>
              </TextRun>
            </TextRuns>
            <Style>
              <TextAlign>Right</TextAlign>
            </Style>
          </Paragraph>
        </Paragraphs>
      </Textbox>
    </CellContents>
  </TablixHeader>
</TablixMember>
</TablixMembers>
</TablixMember>

```

The second dynamic column is grouping on the Order Date month. The [DataSet](#) contains data for only a single month. Therefore, the example renders only a single column for this **dynamic group**. The size of the **TablixHeader** element is equal to the sum of the sizes of the first dynamic column headers. This effectively means that the [CellContents](#) span two rendered header rows. The text box is colored pink.

```

<TablixMember>
  <Group Name="OrdersGroup">
    <GroupExpressions>
      GroupExpression>=Fields!OrderDate.Value.Month()</GroupExpression>
    </GroupExpressions>
  </Group>
  <SortExpressions>
    <SortExpression>
      <Value>=Fields!OrderDate.Value.Month()</Value>
    </SortExpression>
  </SortExpressions>
  <TablixHeader>
    <Size>0.94271in</Size>
    <CellContents>
      <Textbox Name="OrdersColumnHeader">
        <CanGrow>true</CanGrow>
        <KeepTogether>true</KeepTogether>
        <Paragraphs>
          <Paragraph>
            <TextRuns>
              <TextRun>
                <Value>Total Orders</Value>
                <Style>
                  <FontSize>12pt</FontSize>
                  <FontWeight>Bold</FontWeight>
                </Style>
              </TextRun>
            </TextRuns>
            <Style>
              <TextAlign>Right</TextAlign>
            </Style>
          </Paragraph>
        </Paragraphs>
        <Style>
          <Border>
            <Color>LightGrey</Color>
            <Style>Solid</Style>
          </Border>
          <BackgroundColor>Pink</BackgroundColor>
        </Style>
      </Textbox>
    </CellContents>
  </TablixHeader>
</TablixMember>
</TablixMembers>
</TablixColumnHierarchy>

```

The row hierarchy defines a dynamic group (colored cornsilk in the preceding figure) with a child dynamic group (colored yellow in the figure above) and two static groups which act as header and footer to the child dynamic group.

```
<TablixRowHierarchy>
```

The parent **dynamic row** is grouping on ProductCategoryName. The text box is colored cornsilk. This dynamic row is repeated for each unique group item, and the child row groups are repeated within it.

```

<TablixMembers>
  <TablixMember>
    <Group Name="CategoryGroup">
      <GroupExpressions>
        GroupExpression>=Fields!ProductCategoryName.Value</GroupExpression>
      </GroupExpressions>
    </Group>
  </TablixMember>
</TablixMembers>

```

```

<SortExpressions>
  <SortExpression>
    <Value>=Fields!ProductCategoryName.Value</Value>
  </SortExpression>
</SortExpressions>
<TablixHeader>

```

Because the **TablixHeader** element is within the row hierarchy, the size is the width of a column.

```

  <Size>1in</Size>
  <CellContents>
    <Textbox Name="CategoryRowHeader">
      <CanGrow>true</CanGrow>
      <KeepTogether>true</KeepTogether>
      <Paragraphs>
        <Paragraph>
          <TextRuns>
            <TextRun>
              <Value>=Fields!ProductCategoryName.Value</Value>
            </TextRun>
          </TextRuns>
        </Paragraph>
      </Paragraphs>
      <Style>
        <Border>
          <Color>LightGrey</Color>
          <Style>Solid</Style>
        </Border>
        <BackgroundColor>CornSilk</BackgroundColor>
      </Style>
    </Textbox>
  </CellContents>
</TablixHeader>

```

The parent dynamic row has a child static member that is used to display row header text at the beginning of each child dynamic row. The header text is "First order received".

```

<TablixMembers>
  <TablixMember>
    <TablixHeader>
      <Size>1.76562in</Size>
      <CellContents>
        <Textbox Name="ProductHeaderRowHeader">
          <CanGrow>true</CanGrow>
          <KeepTogether>true</KeepTogether>
          <Paragraphs>
            <Paragraph>
              <TextRuns>
                <TextRun>
                  <Value>First order received</Value>
                  <Style>
                    <FontWeight>Bold</FontWeight>
                  </Style>
                </TextRun>
              </TextRuns>
            </Paragraph>
          </Paragraphs>
          <Style>
            <Border>
              <Color>LightGrey</Color>
              <Style>Solid</Style>
            </Border>
          </Style>
        </Textbox>
      </CellContents>
    </TablixHeader>
  </TablixMember>
</TablixMembers>

```

```

    </CellContents>
  </TablixHeader>

```

Specifying the [TablixMember.KeepWithGroup](#) value as "After" means that an attempt is to be made to render this row with the **dynamic member** (in yellow) that follows.

```

    <KeepWithGroup>After</KeepWithGroup>
  </TablixMember>

```

The child dynamic member groups on the ProductName field and will be displayed once for each unique ProductName. The text box is colored yellow.

```

<TablixMember>
  <Group Name="Product">
    <GroupExpressions>
      <GroupExpression>=Fields!ProductName.Value</GroupExpression>
    </GroupExpressions>
  </Group>
  <TablixHeader>
    <Size>1.76562in</Size>
    <CellContents>
      <Textbox Name="ProductRowHeader">
        <CanGrow>true</CanGrow>
        <KeepTogether>true</KeepTogether>
        <Paragraphs>
          <Paragraph>
            <TextRuns>
              <TextRun>
                <Value>=Fields!ProductName.Value</Value>
              </TextRun>
            </TextRuns>
          </Paragraph>
        </Paragraphs>
        <Style>
          <Border>
            <Color>LightGrey</Color>
            <Style>Solid</Style>
          </Border>
          <BackgroundColor>Yellow</BackgroundColor>
        </Style>
      </Textbox>
    </CellContents>
  </TablixHeader>
</TablixMember>

```

The parent dynamic row has a child static member that is used to display row footer text at the end of each child dynamic row. The footer text is "Total".

```

<TablixMember>
  <TablixHeader>
    <Size>1.76562in</Size>
    <CellContents>
      <Textbox Name="ProductFooterRowHeader">
        <CanGrow>true</CanGrow>
        <KeepTogether>true</KeepTogether>
        <Paragraphs>
          <Paragraph>
            <TextRuns>
              <TextRun>
                <Value>Total</Value>
              <Style>

```

```

        <FontWeight>Bold</FontWeight>
    </Style>
</TextRun>
</TextRuns>
</Paragraph>
</Paragraphs>
<Style>
    <Border>
        <Color>LightGrey</Color>
        <Style>Solid</Style>
    </Border>
    <TopBorder>
        <Color>Black</Color>
    </TopBorder>
    <BottomBorder>
        <Color>Black</Color>
        <Width>2pt</Width>
    </BottomBorder>
</Style>
</Textbox>
</CellContents>
</TablixHeader>

```

Specifying the **TablixMember.KeepWithGroup** value as "Before" means that an attempt is to be made to render this row with the preceding dynamic member (in yellow).

```

        <KeepWithGroup>Before</KeepWithGroup>
    </TablixMember>
</TablixMembers>
</TablixMember>
</TablixMembers>
</TablixRowHierarchy>

```

A [TablixCorner](#) element is specified (colored lime in the preceding figure) to occupy the top left corner. Observe that there is no ability to specify width or heights within the **TablixCorner**. The width of the columns is defined by the size of the **TablixHeader** elements in the row hierarchy. The height of the rows is defined by the size of the **TablixHeader** elements in the column hierarchy.

```

<TablixCorner>
    <TablixCornerRows>

```

There are two [TablixCornerRow](#) elements specified. Two are required because of the following rule that is stated in sections 2.23 and [2.37.1](#):

"If a **TablixCorner** element is specified, the quantity of descendant **TablixCornerRow** elements in the tablix MUST equal the quantity of unique cumulative heights for each tablix **column member** of the tablix. 'Cumulative height' for a tablix column member is defined as the sum of the values of all [TablixHeader.Size](#) elements that are descendants of the tablix column member."

Column hierarchy	Size	Cumulative height
Tablix member (SalesPerson)	.43438	.43438
Tablix member "List Price"	.43438	.43438 + .45833 = .94721
Tablix member "Standard Cost"	.43438	.43438 + .45833 = .94721
Tablix member (SalesPerson)	.94721	.94721

### Computation of quantity of TablixCornerRow elements

This table shows the structure of the **TablixHeaders** within the column hierarchy, identifying the size values. The cumulative height calculation shows the addition of the **RdlSize** with all of its parent's **RdlSizes**. There are two unique cumulative heights: these are .48438 and .94721. Therefore, there need to be two **TablixCornerRow** elements.

There are two [TablixCornerCell](#) elements specified in each row. Two are required because of the following rule that is stated in sections 2.23 and [2.38.1](#):

"If a **TablixCornerRow** element is specified, the quantity of descendant **TablixCornerCell** elements in the tablix MUST equal the quantity of unique cumulative widths for each tablix **row member** of the tablix. 'Cumulative width' for a tablix row member is defined as the sum of the values of all **TablixHeader.Size** elements that are descendants of the tablix row member."

Row hierarchy	Size	Cumulative width
Tablix member (ProductCategoryName)	1.00000	1.00000
Tablix member "First order received"	1.76562	1.00000 + 1.76562= 2.76562
Tablix member (ProductName)	1.76562	1.00000 + 1.76562= 2.76562
Tablix member "Total"	1.76562	1.00000 + 1.76562= 2.76562

### Tablix example 3—computation of quantity of **TablixCornerCell** elements

The preceding table shows the structure of the **TablixHeaders** within the row hierarchy, identifying the size values. The cumulative width calculation shows the addition of the size with all of its parent's sizes. There are two unique cumulative heights: these are 1.00000 and 1.76562. Therefore, there need to be two **TablixCornerCell** elements in each **TablixCornerRow** element.

```

<TablixCornerRow>
  <TablixCornerCell>
    <CellContents>
      <Textbox Name="Textbox4">
        <CanGrow>true</CanGrow>
        <KeepTogether>true</KeepTogether>
        <Paragraphs>
          <Paragraph>
            <TextRuns>
              <TextRun>
                <Value>Bicycle sales for June 2004</Value>
                <Style>
                  <FontSize>12pt</FontSize>
                  <FontWeight>Bold</FontWeight>
                  <TextDecoration>Underline</TextDecoration>
                </Style>
              </TextRun>
            </TextRuns>
          </Paragraph>
        </Paragraphs>
        <Style>
          <Border>
            <Color>LightGrey</Color>
            <Style>Solid</Style>
          </Border>
          <BackgroundColor>Lime</BackgroundColor>
        </Style>
      </Textbox>
    </CellContents>
  </TablixCornerCell>
</TablixCornerRow>

```

The first row of the corner has a title that uses two cells. The **CellContents** specifies a **ColSpan** of 2 for the first **TablixCornerCell** and the second **TablixCornerCell** is specified as empty.



```

        <ColSpan>2</ColSpan>
    </CellContents>
</TablixCornerCell>
<TablixCornerCell />
</TablixCornerRow>

```

The second row of the corner contains two text boxes to display "Category" and "Product Name" as column titles over the row headers.

```

<TablixCornerRow>
  <TablixCornerCell>
    <CellContents>
      <Textbox Name="Textbox18">
        <CanGrow>true</CanGrow>
        <KeepTogether>true</KeepTogether>
        <Paragraphs>
          <Paragraph>
            <TextRuns>
              <TextRun>
                <Value>Category</Value>
                <Style>
                  <FontWeight>Bold</FontWeight>
                </Style>
              </TextRun>
            </TextRuns>
          </Paragraph>
        </Paragraphs>
        <Style>
          <Border>
            <Color>LightGrey</Color>
            <Style>Solid</Style>
          </Border>
          <BackgroundColor>Lime</BackgroundColor>
        </Style>
      </Textbox>
    </CellContents>
  </TablixCornerCell>
  <TablixCornerCell>
    <CellContents>
      <Textbox Name="Textbox7">
        <CanGrow>true</CanGrow>
        <KeepTogether>true</KeepTogether>
        <Paragraphs>
          <Paragraph>
            <TextRuns>
              <TextRun>
                <Value>Product Name</Value>
                <Style>
                  <FontWeight>Bold</FontWeight>
                </Style>
              </TextRun>
            </TextRuns>
          </Paragraph>
        </Paragraphs>
        <Style>
          <Border>
            <Color>LightGrey</Color>
            <Style>Solid</Style>
          </Border>
          <BackgroundColor>Lime</BackgroundColor>
        </Style>
      </Textbox>
    </CellContents>
  </TablixCornerCell>
</TablixCornerRow>
</TablixCornerRows>

```

```
</TablixCorner>
```

The body of the tablix defines the cells that can repeat for both columns and rows. The widths of each [TablixColumn](#) specify the width of each column as it repeats.

```
<TablixBody>
  <TablixColumns>
    <TablixColumn>
      <Width>1.47396in</Width>
    </TablixColumn>
    <TablixColumn>
      <Width>1.29167in</Width>
    </TablixColumn>
    <TablixColumn>
      <Width>1in</Width>
    </TablixColumn>
  </TablixColumns>
  <TablixRows>
    <TablixRow>
```

The heights of each [TablixRow](#) specify the height of each row as it repeats.

```
    <Height>0.25in</Height>
  <TablixCells>
    <TablixCell>
      <CellContents>
        <Textbox Name="OrderDate">
          <CanGrow>true</CanGrow>
          <KeepTogether>true</KeepTogether>
          <Paragraphs>
            <Paragraph>
              <TextRuns>
                <TextRun>
```

Because the cells are being used within groups, the data expression uses **aggregate functions**.

```
                <Value>=First(Fields!OrderDate.Value)</Value>
                <Style>
                  <Format>MMM dd, yyyy</Format>
                </Style>
              </TextRun>
            </Paragraph>
          </TextRuns>
        <Style>
          <TextAlign>Center</TextAlign>
        </Style>
      </Paragraph>
    </Paragraphs>
  <Style>
    <Border>
      <Color>LightGrey</Color>
      <Style>Solid</Style>
    </Border>
  </Style>
</Textbox>
```

The **CellContents** for the [TablixCell](#) spreads across two columns using a **ColSpan** element. This means that the following **TablixCell** needs to be specified as empty.

```
    <ColSpan>2</ColSpan>
  </CellContents>
```

```

</TablixCell>
<TablixCell />
<TablixCell>
  <CellContents>
    <Textbox Name="Textbox40">
      <CanGrow>true</CanGrow>
      <KeepTogether>true</KeepTogether>
      <Paragraphs>
        <Paragraph>
          <TextRuns>
            <TextRun>
              <Value />
            </TextRun>
          </TextRuns>
          <Style>
            <TextAlign>Center</TextAlign>
          </Style>
        </Paragraph>
      </Paragraphs>
      <Style>
        <Border>
          <Color>LightGrey</Color>
          <Style>Solid</Style>
        </Border>
        <BackgroundColor>Pink</BackgroundColor>
      </Style>
    </Textbox>
  </CellContents>
</TablixCell>
</TablixCells>
</TablixRow>
<TablixRow>
  <Height>0.39062in</Height>
  <TablixCells>
    <TablixCell>
      <CellContents>
        <Textbox Name="ListPrice">
          <CanGrow>true</CanGrow>
          <KeepTogether>true</KeepTogether>
          <Paragraphs>
            <Paragraph>
              <TextRuns>
                <TextRun>

```

The cell is associated with the Product dynamic group. The aggregate function will apply across the same **scope**.

```

          <Value>=Sum(Fields!ListPrice.Value)</Value>
        </TextRun>
      </TextRuns>
    </Paragraph>
  </Paragraphs>
  <Style>
    <Border>
      <Color>LightGrey</Color>
      <Style>Solid</Style>
    </Border>
  </Style>
</Textbox>
</CellContents>
</TablixCell>
<TablixCell>
  <CellContents>
    <Textbox Name="StandardCost">
      <CanGrow>true</CanGrow>
      <KeepTogether>true</KeepTogether>
      <Paragraphs>

```

```

        <Paragraph>
          <TextRuns>
            <TextRun>
              <Value>=Sum(Fields!StandardCost.Value)</Value>
            </TextRun>
          </TextRuns>
        </Paragraph>
      </Paragraphs>
    </CellContents>
  </TablixCell>
</TablixCell>
<CellContents>
  <Textbox Name="OrderQty">
    <CanGrow>true</CanGrow>
    <KeepTogether>true</KeepTogether>
    <Paragraphs>
      <Paragraph>
        <TextRuns>
          <TextRun>
            <Value>=Sum(Fields!OrderQty.Value)</Value>
          </TextRun>
        </TextRuns>
      </Paragraph>
    </Paragraphs>
    <Style>
      <Border>
        <Color>LightGrey</Color>
        <Style>Solid</Style>
      </Border>
      <BackgroundColor>Pink</BackgroundColor>
    </Style>
  </Textbox>
</CellContents>
</TablixCell>
</TablixCells>
</TablixRow>
<TablixRow>
  <Height>0.25in</Height>
  <TablixCells>
    <TablixCell>
      <CellContents>
        <Textbox Name="ListPrice1">
          <CanGrow>true</CanGrow>
          <KeepTogether>true</KeepTogether>
          <Paragraphs>
            <Paragraph>
              <TextRuns>
                <TextRun>

```

This cell is associated with the static group under the dynamic grouping of Product Category. The aggregate will apply to Product Category.

```

          <Value>=Sum(Fields!ListPrice.Value)</Value>
        </TextRun>
      </TextRuns>
    </Paragraph>
  </Paragraphs>
  <Style>
    <Border>
      <Color>LightGrey</Color>

```

```

        <Style>Solid</Style>
    </Border>
    <TopBorder>
        <Color>Black</Color>
    </TopBorder>
    <BottomBorder>
        <Color>Black</Color>
        <Width>2pt</Width>
    </BottomBorder>
</Style>
</Textbox>
</CellContents>
</TablixCell>
<TablixCell>
    <CellContents>
        <Textbox Name="StandardCost1">
            <CanGrow>true</CanGrow>
            <KeepTogether>true</KeepTogether>
            <Paragraphs>
                <Paragraph>
                    <TextRuns>
                        <TextRun>
                            <Value>=Sum(Fields!StandardCost.Value)</Value>
                        </TextRun>
                    </TextRuns>
                </Paragraph>
            </Paragraphs>
            <Style>
                <Border>
                    <Color>LightGrey</Color>
                    <Style>Solid</Style>
                </Border>
                <TopBorder>
                    <Color>Black</Color>
                </TopBorder>
                <BottomBorder>
                    <Color>Black</Color>
                    <Width>2pt</Width>
                </BottomBorder>
            </Style>
        </Textbox>
    </CellContents>
</TablixCell>
<TablixCell>
    <CellContents>
        <Textbox Name="Textbox42">
            <CanGrow>true</CanGrow>
            <KeepTogether>true</KeepTogether>
            <Paragraphs>
                <Paragraph>
                    <TextRuns>
                        <TextRun>
                            <Value>=Sum(Fields!OrderQty.Value)</Value>
                        </TextRun>
                    </TextRuns>
                </Paragraph>
            </Paragraphs>
            <Style>
                <Border>
                    <Color>LightGrey</Color>
                    <Style>Solid</Style>
                </Border>
                <TopBorder>
                    <Color>Black</Color>
                </TopBorder>
                <BottomBorder>
                    <Color>Black</Color>
                    <Width>2pt</Width>
                </BottomBorder>
            </Style>
        </Textbox>
    </CellContents>
</TablixCell>

```

```

        <BackgroundColor>Pink</BackgroundColor>
    </Style>
</Textbox>
</CellContents>
</TablixCell>
</TablixCells>
</TablixRow>
</TablixRows>
</TablixBody>
<DataSetName>DataSet1</DataSetName>
<Style>
    <Border>
        <Style>None</Style>
    </Border>
</Style>
</Tablix>

```

### 3.10 Data

The following is an example that shows how to define the data that will be used for the [Report](#) element. The example shows a [DataSource](#) and [DataSet](#) definition, including the [Query](#), which has [QueryParameters](#) that reference [ReportParameters](#), and the [Fields](#) for the **DataSet**, with one field being calculated based on two other fields in the **Fields** collection.

```

1 <DataSources>
2   <DataSource Name="DataSource1">
3     <ConnectionProperties>
4       <DataProvider>SQL</DataProvider>
5       <ConnectString>data source=dataServer; initial
6         catalog=northwind;</ConnectString>
7     </ConnectionProperties>
8   </DataSource>
9</DataSources>
10<DataSets>
11 <DataSet Name="DataSet1">
12   <Fields>
13     <Field Name="ProductID">
14       <DataField>ProductID</DataField>
15     </Field>
16     <Field Name="ProductName">
17       <DataField>ProductName</DataField>
18     </Field>
19     <Field Name="SupplierID">
20       <DataField>SupplierID</DataField>
21     </Field>
22     <Field Name="CategoryID">
23       <DataField>CategoryID</DataField>
24     </Field>
25     <Field Name="QuantityPerUnit">
26       <DataField>QuantityPerUnit</DataField>
27     </Field>
28     <Field Name="UnitPrice">
29       <DataField>UnitPrice</DataField>
30     </Field>
31     <Field Name="UnitsInStock">
32       <DataField>UnitsInStock</DataField>
33     </Field>
34     <Field Name="UnitsOnOrder">
35       <DataField>UnitsOnOrder</DataField>
36     </Field>
37     <Field Name="ReorderLevel">
38       <DataField>ReorderLevel</DataField>
39     </Field>
40     <Field Name="Discontinued">

```

```

41     <DataField>Discontinued</DataField>
42   </Field>
43   <Field Name="TotalPrice">
44     <Value>=Fields!UnitsInStock.Value * Fields!UnitPrice.Value</Value>
45   </Field>
46 </Fields>
47 <Query>
48   <DataSourceName>DataSource1</DataSourceName>
49   <CommandText>SELECT * From Products WHERE UnitsInStock >= @UnitsInStockMin
        AND UnitsInStock <= @UnitsInStockMax;</CommandText>
50   <QueryParameters>
51     <QueryParameter Name="@UnitsInStockMin">
52       <Value>=Parameters!UnitsInStockMin.Value</Value>
53     </QueryParameter>
54     <QueryParameter Name="@UnitsInStockMax">
55       <Value>=Parameters!UnitsInStockMax.Value</Value>
56     </QueryParameter>
57   </QueryParameters>
58 </Query>
59 </DataSet>
60</DataSets>

```

### 3.11 Subreport

The following RDL definition shows an example of a [Subreport](#) definition that would be used within a **data region**. The example includes specifying the location of the **report definition** to use for the subreport by specifying the path by using the [Subreport.ReportName](#) element. The report being referenced requires three **report parameters** to be supplied with values, which are specified by using the [Subreport.Parameters](#) element.

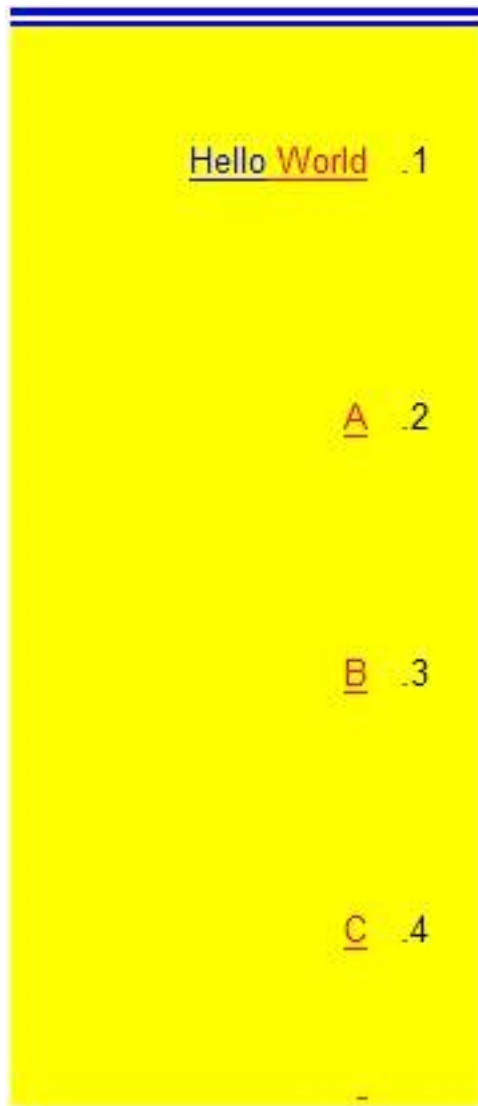
```

<Subreport Name="Subreport1">
  <ReportName>/Subreports/CustomerOrder.rdl</ReportName>
  <Parameters>
    <Parameter Name="CustomerID">
      <Value>=Fields!CustomerID.Value</Value>
    </Parameter>
    <Parameter Name="OrderID">
      <Value>=Fields!OrderID.Value</Value>
    </Parameter>
    <Parameter Name="ShowDetails">
      <Value>=(Count(Fields!OrderID.Value) < 5)</Value>
    </Parameter>
  </Parameters>
  <Top>0.69667in</Top>
  <Left>0.43625in</Left>
  <Height>2.61458in</Height>
  <Width>4.77083in</Width>
  <ZIndex>1</ZIndex>
</Subreport>

```

### 3.12 TextBox

The following is an example **text box** that was created by using the [Textbox](#) element.



**Figure 20: Text box example**

The [Name](#) attribute of the **Textbox** element is set to "Textbox1". "Textbox1" has its [Textbox.CanGrow](#) element set to true. This means that, although the [Textbox.Height](#) element is set to only 0.72917 inches, extra spacing or padding introduced by other elements within "Textbox1" can make "Textbox1" grow vertically beyond 0.72917 inches. "Textbox1" contains four [Paragraph](#) elements within its [Paragraphs](#) collection.

```
<Textbox Name="Textbox1">  
  <CanGrow>true</CanGrow>  
  <KeepTogether>true</KeepTogether>  
  <Paragraphs>
```

The first **Paragraph** in the **Paragraphs** collection for "Textbox1" contains two [TextRun](#) elements, which correspond separately to the words "Hello" and "World". The **TextRun** element that has [TextRun.Value](#) set to "Hello" has [Style.Color](#) set to "Blue". The **TextRun** element that has [TextRun.Value](#) set to "World" has [Style.Color](#) set to "Red".



```

<Paragraph>
  <TextRuns>
    <TextRun>
      <Value>Hello</Value>
      <Style>
        <Color>Blue</Color>
      </Style>
    </TextRun>
    <TextRun>
      <Value> World</Value>
      <Style>
        <Color>Red</Color>
      </Style>
    </TextRun>
  </TextRuns>
</ListStyle>Numbered</ListStyle>
<ListLevel>1</ListLevel>

```

Because [Style.LineHeight](#) is set to "1in", each **Paragraph** within "Textbox1" takes up at least one inch of vertical space. Because **Textbox.CanGrow** is set to true, "Textbox1" actually is much larger in height than its specified **Textbox.Height** value of 0.72971 inches.

```

  <Style>
    <LineHeight>1in</LineHeight>
  </Style>
</Paragraph>
<Paragraph>
  <TextRuns>
    <TextRun>
      <Value>A</Value>
      <Style>
        <Color>Red</Color>
      </Style>
    </TextRun>
  </TextRuns>
</ListStyle>Numbered</ListStyle>
<ListLevel>1</ListLevel>
<Style>
  <LineHeight>1in</LineHeight>
</Style>
</Paragraph>
<Paragraph>
  <TextRuns>
    <TextRun>
      <Value>B</Value>
      <Style>
        <Color>Red</Color>
      </Style>
    </TextRun>
  </TextRuns>
</ListStyle>Numbered</ListStyle>
<ListLevel>1</ListLevel>
<Style>
  <LineHeight>1in</LineHeight>
</Style>
</Paragraph>
<Paragraph>
  <TextRuns>
    <TextRun>
      <Value>C</Value>
      <Style>
        <Color>Red</Color>
      </Style>
    </TextRun>
  </TextRuns>
</ListStyle>Numbered</ListStyle>

```

```

    <ListLevel>1</ListLevel>
    <Style>
      <LineHeight>1in</LineHeight>
    </Style>
  </Paragraph>
  <Paragraph>
    <TextRuns>
      <TextRun>
        <Value />
        <Style>
          <Color>Red</Color>
        </Style>
      </TextRun>
    </TextRuns>
    <ListLevel>1</ListLevel>
    <Style />
  </Paragraph>
</Paragraphs>
<rd:DefaultName>Textbox1</rd:DefaultName>
<Top>0.85292in</Top>
<Left>2.19667in</Left>
<Height>0.72917in</Height>
<Width>1.84375in</Width>

```

"Textbox1" has a [Style.TopBorder](#) element with [Border.Style](#) set to "Blue", **Border.Style** set to "Double", and [Border.Width](#) set to "5pt". Because these properties are set only for **Style.TopBorder**, and no other [Style](#) border elements or a generic [Style.Border](#) element, the left, right, and bottom sides of "Textbox1" are not affected. This is shown in the figure at the beginning of this section. Additionally, "Textbox1" has [Style.Direction](#) set to "RTL", which is why the numbering in the list within "Textbox1" occurs to the right of the actual text.

```

<Style>
  <Border>
    <Style>None</Style>
  </Border>
  <TopBorder>
    <Color>Blue</Color>
    <Style>Double</Style>
    <Width>5pt</Width>
  </TopBorder>
  <BackgroundColor>Yellow</BackgroundColor>
  <TextAlign>Center</TextAlign>
  <PaddingLeft>2pt</PaddingLeft>
  <PaddingRight>2pt</PaddingRight>
  <PaddingTop>2pt</PaddingTop>
  <PaddingBottom>2pt</PaddingBottom>
  <Direction>RTL</Direction>
</Style>
</Textbox>

```

### 3.13 GaugePanel

The following is an example RDL definition of a [GaugePanel](#) element that is used to create the **gauge panel** shown in the following figure. The **GaugePanel** element has its [Name](#) attribute set to "GaugePanel3". "GaugePanel3" has the [GaugePanel.DataSetName](#) element set to "DataSet1". The [DataSet](#) named "DataSet1" exposes two [Fields](#) named "UnitsInStock" and "UnitsOnOrder" for "GaugePanel3" to use.



**Figure 21: Gauge panel with single radial gauge**

The gauge panel contains a single [RadialGauges](#) collection. This **RadialGauges** collection contains only one [RadialGauge](#) element. This **RadialGauge** element has its **Name** attribute set to "RadialGauge1".

```
<GaugePanel Name="GaugePanel13">
  <RadialGauges>
    <RadialGauge Name="RadialGauge1">
```

"RadialGauge1" contains only one [RadialScale](#) element within its [RadialGauge.GaugeScales](#) collection. This **RadialScale** element has its **Name** attribute set to "RadialScale1".

```
<GaugeScales>
  <RadialScale Name="RadialScale1">
    <GaugePointers>
```

"RadialScale1" contains only one [RadialPointer](#) element within its [RadialScale.GaugePointers](#) collection. This **RadialPointer** element has its **Name** attribute set to "RadialPointer1". "RadialPointer1" has [RadialPointer.GaugeInputValue](#) set to "Sum(Fields!UnitsOnOrder.Value)", which means that the pointer will point at the value that is equal to the cumulative sum of all values for the "UnitsOnOrder" [Field](#), which evaluates to a value between 500 and 1000.

```
<RadialPointer Name="RadialPointer1">
  <PointerCap>
    <Style>
      <BackgroundColor>White</BackgroundColor>
      <BackgroundGradientType>DiagonalLeft</BackgroundGradientType>
      <BackgroundGradientEndColor>DimGray</BackgroundGradientEndColor>
      <BackgroundHatchType>None</BackgroundHatchType>
    </Style>
    <CapImage>
      <OffsetX>0pt</OffsetX>
      <OffsetY>0pt</OffsetY>
      <Source>External</Source>
      <Value />
    </CapImage>
    <OnTop>true</OnTop>
    <CapStyle>RoundedWithWideIndentation</CapStyle>
    <Width>33</Width>
  </PointerCap>
  <NeedleStyle>Tapered</NeedleStyle>
```

```

<Style>
  <Border>
    <Style>None</Style>
  </Border>
  <BackgroundColor>White</BackgroundColor>
  <BackgroundGradientType>DiagonalLeft</BackgroundGradientType>
  <BackgroundGradientEndColor>Orange</BackgroundGradientEndColor>
  <BackgroundHatchType>None</BackgroundHatchType>
  <ShadowOffset>1pt</ShadowOffset>
</Style>
<GaugeInputValue>
  <Value>=Sum(Fields!UnitsOnOrder.Value)</Value>
  <Multiplier>1</Multiplier>
</GaugeInputValue>
<PointerImage>
  <OffsetX>0pt</OffsetX>
  <OffsetY>0pt</OffsetY>
  <Source>External</Source>
  <Value />
</PointerImage>
<MarkerLength>25</MarkerLength>
<MarkerStyle>Wedge</MarkerStyle>
<Placement>Inside</Placement>
<Width>15</Width>
</RadialPointer>
</GaugePointers>

```

"RadialScale1" contains only one [ScaleRange](#) element within its [ScaleRanges](#) collection. This **ScaleRange** element has its **Name** attribute set to "RadialRange1". "RadialRange1" has ScaleRange's [Style.BackgroundGradientEndColor](#) set to "Salmon" and ScaleRange's [Style.BackgroundGradientType](#) set to "DiagonalLeft". These settings affect the viewing layout of "RadialRange1", as seen in the previous figure. The width of "RadialRange1" varies from 15 percent of the radius of "RadialScale1" at the beginning of the range to 30 percent of the radius of "RadialScale1" at the end of the range. The actual values for "RadialRange1" go from 10 percent to 90 percent of the cumulative sum of all values for the "UnitsInStock" field.

```

<ScaleRanges>
  <ScaleRange Name="RadialRange1">
    <Style>
      <Border>
        <Color>Silver</Color>
        <Style>Solid</Style>
      </Border>
      <BackgroundColor>#00ffffff</BackgroundColor>
      <BackgroundGradientType>DiagonalLeft</BackgroundGradientType>
      <BackgroundGradientEndColor>Salmon</BackgroundGradientEndColor>
      <BackgroundHatchType>None</BackgroundHatchType>
      <ShadowOffset>0pt</ShadowOffset>
    </Style>
    <DistanceFromScale>30</DistanceFromScale>
    <StartValue>
      <Value>=0.1*Sum(Fields!UnitsInStock.Value)</Value>
      <Multiplier>1</Multiplier>
    </StartValue>
    <EndValue>
      <Value>=0.9*Sum(Fields!UnitsInStock.Value)</Value>
      <Multiplier>1</Multiplier>
    </EndValue>
    <StartWidth>15</StartWidth>
    <EndWidth>30</EndWidth>
    <Placement>Inside</Placement>
  </ScaleRange>
</ScaleRanges>
<Style>
  <Border>

```

```

        <Style>None</Style>
    </Border>
    <BackgroundColor>CornflowerBlue</BackgroundColor>
    <BackgroundGradientType>None</BackgroundGradientType>
    <BackgroundGradientEndColor>White</BackgroundGradientEndColor>
    <BackgroundHatchType>None</BackgroundHatchType>
    <ShadowOffset>0pt</ShadowOffset>
</Style>
<Interval>NaN</Interval>
<IntervalOffset>NaN</IntervalOffset>

```

"RadialScale1" has [RadialScale.MinimumValue](#) set to 0 and [RadialScale.MaximumValue](#) set to "=Sum(Fields!UnitsInStock.Value)". This means that the scale that is represented by "RadialScale1" can go from 0 to the cumulative total of all product units that are in stock as recorded by "DataSet1". This cumulative total is slightly greater than 3000.

```

    <MaximumValue>
        <Value>=Sum(Fields!UnitsInStock.Value)</Value>
        <Multiplier>1</Multiplier>
    </MaximumValue>
    <MinimumValue>
        <Value>0</Value>
        <Multiplier>1</Multiplier>
    </MinimumValue>

```

The [RadialScale.GaugeMajorTickMarks](#) has [TickMarkStyle.Placement](#) set to "Outside" and no [TickMarkStyle.Shape](#) set. Therefore, the major tick marks, with their intervals defaulted to 500, are placed outside of "RadialScale1" and contain a slim (because [TickMarkStyle.Width](#) is set to only 2) rectangular shape.

```

<GaugeMajorTickMarks>
    <Interval>NaN</Interval>
    <IntervalOffset>NaN</IntervalOffset>
    <Style>
        <Border>
            <Color>DimGray</Color>
            <Style>None</Style>
        </Border>
        <BackgroundColor>DimGray</BackgroundColor>
    </Style>
    <Placement>Outside</Placement>
    <TickMarkImage>
        <Source>External</Source>
        <Value />
    </TickMarkImage>
    <Length>18</Length>
    <Width>2</Width>
</GaugeMajorTickMarks>

```

The [RadialScale.GaugeMinorTickMarks](#) has **TickMarkStyle.Placement** set to "Cross" and no **TickMarkStyle.Shape** set. Therefore, the minor tick marks, with their intervals defaulted to 100, are placed directly across "RadialScale1" and contain a slim (because **TickMarkStyle.Width** is set to only 1) rectangular shape.

```

<GaugeMinorTickMarks>
    <Interval>NaN</Interval>
    <IntervalOffset>NaN</IntervalOffset>
    <Style>
        <Border>
            <Color>DimGray</Color>
            <Style>None</Style>

```

```

        </Border>
        <BackgroundColor>DimGray</BackgroundColor>
    </Style>
    <Placement>Cross</Placement>
    <TickMarkImage>
        <Source>External</Source>
        <Value />
    </TickMarkImage>
    <Length>8</Length>
    <Width>1</Width>
</GaugeMinorTickMarks>
<MaximumPin>
    <PinLabel>
        <Style>
            <FontSize>12pt</FontSize>
        </Style>
        <DistanceFromScale>0</DistanceFromScale>
    </PinLabel>
    <Style>
        <Border>
            <Color>DimGray</Color>
            <Style>Solid</Style>
        </Border>
        <BackgroundColor>WhiteSmoke</BackgroundColor>
    </Style>
    <Placement>Cross</Placement>
    <TickMarkImage>
        <Source>External</Source>
        <Value />
    </TickMarkImage>
    <Length>6</Length>
    <Width>6</Width>
    <Shape>Circle</Shape>
    <Hidden>true</Hidden>
</MaximumPin>
<MinimumPin>
    <PinLabel>
        <Style>
            <FontSize>12pt</FontSize>
        </Style>
        <DistanceFromScale>0</DistanceFromScale>
    </PinLabel>
    <Style>
        <Border>
            <Color>DimGray</Color>
            <Style>Solid</Style>
        </Border>
        <BackgroundColor>WhiteSmoke</BackgroundColor>
    </Style>
    <Placement>Cross</Placement>
    <TickMarkImage>
        <Source>External</Source>
        <Value />
    </TickMarkImage>
    <Length>6</Length>
    <Width>6</Width>
    <Shape>Circle</Shape>
    <Hidden>true</Hidden>
</MinimumPin>
<ScaleLabels>
    <Style>
        <FontSize>14pt</FontSize>
        <Color>#404040</Color>
    </Style>
    <Interval>NaN</Interval>
    <IntervalOffset>NaN</IntervalOffset>
    <DistanceFromScale>0</DistanceFromScale>
    <RotateLabels>true</RotateLabels>
    <ShowEndLabels>true</ShowEndLabels>

```

```

        <UseFontPercent>true</UseFontPercent>
    </ScaleLabels>
    <Width>0</Width>
</RadialScale>
</GaugeScales>

```

"RadialGauge1" contains a [RadialGauge.BackFrame](#) element. For the [BackFrame](#), [BackFrame.FrameShape](#) is set to "CustomCircular2", which defines the frame shape of "RadialGauge1", as shown in the figure.

```

<BackFrame>
  <Style>
    <Border>
      <Style>None</Style>
    </Border>
    <BackgroundColor>WhiteSmoke</BackgroundColor>
    <BackgroundGradientType>DiagonalLeft</BackgroundGradientType>
    <BackgroundGradientEndColor>Gainsboro</BackgroundGradientEndColor>
    <BackgroundHatchType>None</BackgroundHatchType>
    <ShadowOffset>0pt</ShadowOffset>
  </Style>
  <FrameStyle>Edged</FrameStyle>
  <FrameShape>CustomCircular2</FrameShape>
  <FrameBackground>
    <Style>
      <BackgroundColor>WhiteSmoke</BackgroundColor>
      <BackgroundGradientType>DiagonalLeft</BackgroundGradientType>
      <BackgroundGradientEndColor>Gray</BackgroundGradientEndColor>
      <BackgroundHatchType>None</BackgroundHatchType>
    </Style>
  </FrameBackground>
  <FrameImage>
    <Source>External</Source>
    <Value />
  </FrameImage>
</BackFrame>
<ClipContent>true</ClipContent>
<TopImage>
  <Source>External</Source>
  <Value />
</TopImage>
<AspectRatio>1</AspectRatio>
<Height>100</Height>
<Width>100</Width>
</RadialGauge>
</RadialGauges>

```

"GaugePanel3" also contains one [GaugeLabel](#) element within its [GaugeLabels](#) collection. This [GaugeLabel](#) element has its **Name** attribute set to "GaugeLabel1". "GaugeLabel1" has [GaugeLabel.Text](#) set to "Orders vs. Stock", [Style.Color](#) set to "Red", and [Style.FontWeight](#) set to "Bold", the results of which can be seen in the figure.

```

<GaugeLabels>
  <GaugeLabel Name="GaugeLabel1">
    <Text>Orders vs. Stock</Text>
    <TextShadowOffset>0pt</TextShadowOffset>
    <Style>
      <Border>
        <Style>None</Style>
      </Border>
      <BackgroundColor>#00ffffff</BackgroundColor>
      <BackgroundGradientType>None</BackgroundGradientType>
      <BackgroundGradientEndColor>#00ffffff</BackgroundGradientEndColor>
      <FontWeight>Bold</FontWeight>
    </Style>
  </GaugeLabel>
</GaugeLabels>

```

```

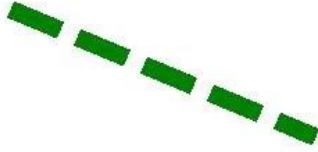
        <Color>Red</Color>
        <ShadowOffset>0pt</ShadowOffset>
    </Style>
    <Top>30.418249130249</Top>
    <Left>40.2298812866211</Left>
    <Height>9</Height>
    <Width>21</Width>
</GaugeLabel>
</GaugeLabels>
<AutoLayout>true</AutoLayout>
<BackFrame>
    <Style>
        <Border>
            <Style>None</Style>
        </Border>
        <BackgroundColor>Gainsboro</BackgroundColor>
        <BackgroundGradientType>DiagonalLeft</BackgroundGradientType>
        <BackgroundGradientEndColor>Gray</BackgroundGradientEndColor>
        <BackgroundHatchType>None</BackgroundHatchType>
        <ShadowOffset>0pt</ShadowOffset>
    </Style>
    <FrameBackground>
        <Style>
            <BackgroundColor>Silver</BackgroundColor>
            <BackgroundGradientType>DiagonalLeft</BackgroundGradientType>
            <BackgroundGradientEndColor>Gray</BackgroundGradientEndColor>
            <BackgroundHatchType>None</BackgroundHatchType>
        </Style>
    </FrameBackground>
    <FrameImage>
        <Source>External</Source>
        <Value />
    </FrameImage>
</BackFrame>
<TopImage>
    <Source>External</Source>
    <Value />
</TopImage>
<DataSetName>DataSet1</DataSetName>
<Top>0.47792in</Top>
<Left>0.62375in</Left>
<Height>2.75in</Height>
<Width>4.54167in</Width>
<Style>
    <Border>
        <Style>None</Style>
    </Border>
    <BackgroundColor>White</BackgroundColor>
</Style>
</GaugePanel>

```

### 3.14 Line

The following example demonstrates an RDL [Line](#) element, which is shown in the following figure.





**Figure 22: Line example**

The [Line.Name](#) is set to "Line1". However, [Line.Top](#) is set to 1.25917in and [Line.Left](#) is set to 4in; [Line.Height](#) is set to "-0.95833in" and [Line.Width](#) is set to -2.29167in. This means that "Line1" starts from the lower-right end and goes to the upper-left end. The [Style.Border](#) element, along with its child elements, controls the appearance of "Line1". [Border.Color](#) is set to Green, [Border.Style](#) is set to "Dashed", and [Border.Width](#) is set to 10pt. These settings are reflected in the preceding figure.

```
<Line Name="Line1">
  <Top>1.25917in</Top>
  <Left>4in</Left>
  <Height>-0.95833in</Height>
  <Width>-2.29167in</Width>
  <Style>
    <Border>
      <Color>Green</Color>
      <Style>Dashed</Style>
      <Width>10pt</Width>
    </Border>
  </Style>
</Line>
```

### 3.15 List

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The following example of a [List](#) element is an extract from an RDL specification of a report. This example is from a report for the AdventureWorks database. It shows a count of how many distinct products there are and the total quantity of products in inventory at different locations within the company in a list. This example illustrates the use of a **list data region** and how to group details by company location.

The following figure shows the **List** example. An instance of the list consists of the label "Location:" followed by the name of the location, the label "Number of Different Products:" followed by a count of the number of different products at that location, and the label "Total Quantity:" followed by the sum total number of products at that location. These six items appear as [Textbox](#) report items placed in free form within the boundaries of a list instance; that is, they are not aligned in any particular rows and columns. They can be placed anywhere within the list instance boundary. The background colors of two of the **Textbox** report items are set to LemonChiffon and LightGreen.

Location: Debur and Polish		Total Quantity:	958
Number of Different Products:	7		
Location: Final Assembly		Total Quantity:	20419
Number of Different Products:	141		
Location: Finished Goods Storage		Total Quantity:	17319
Number of Different Products:	151		
Location: Frame Forming		Total Quantity:	13584
Number of Different Products:	46		
Location: Frame Welding		Total Quantity:	5165
Number of Different Products:	20		
Location: Metal Storage		Total Quantity:	20295
Number of Different Products:	51		
Location: Miscellaneous Storage		Total Quantity:	83173
Number of Different Products:	201		
Location: Paint		Total Quantity:	508
Number of Different Products:	9		
Location: Paint Shop		Total Quantity:	186
Number of Different Products:	5		
Location: Paint Storage		Total Quantity:	110
Number of Different Products:	5		
Location: Sheet Metal Racks		Total Quantity:	5549
Number of Different Products:	12		
Location: Specialized Paint		Total Quantity:	332
Number of Different Products:	3		
Location: Subassembly		Total Quantity:	95477
Number of Different Products:	251		
Location: Tool Crib		Total Quantity:	72899
Number of Different Products:	167		

**Figure 23: Example of a list with grouping**

The following is the extract from the RDL report file that produces the preceding list. The **List** element has its [Name](#) attribute set to "list1". There is a [Sorting](#) element that specifies that this group is sorted by the [Field](#) "Name" in ascending order. As shown in the preceding figure, the locations listed are Debur and Polish, Final Assembly, Finished Goods Storage, and so on, which are sorted in alphabetical (ascending) order. The [Left](#) element specifies that the list is 0.125 inches from the left edge of the body of the report. The list is bound to the data in the [DataSet](#) "DataSet1" by the [List.DataSetName](#) element.

```

<List Name="list1">
  <Sorting>
    <SortBy>
      <SortExpression>=Fields!Name.Value</SortExpression>
      <Direction>Ascending</Direction>
    </SortBy>
  </Sorting>
  <Left>0.125in</Left>
  <DataSetName>DataSet1</DataSetName>

```

The [ReportItems](#) element specifies the six text boxes that comprise a single instance of list data. The text boxes that have the [Name](#) attribute values "textbox3", "textbox4", and "textbox5" are the labels before the data. The text box that has the **Name** attribute value "Name" is the name of the location in the company, and its [Style.BackgroundColor](#) element is specified as LemonChiffon. The text box that has the **Name** attribute value "textbox2" is the total quantity of products in this location in the company, and its [Style.BackgroundColor](#) element is specified as LightGreen. Note that its [Value](#) element is an expression that uses the [Sum](#) aggregate function. The text box that has the **Name** attribute value "textbox1" is the count of distinct products in this location in the company. Note that its **Value** element is an expression that uses the [Count](#) aggregate function.

```

<ReportItems>
  <Textbox Name="textbox1">
    <Top>0.25in</Top>
    <Width>0.5in</Width>
    <Style>
      <TextAlign>Right</TextAlign>
      <PaddingLeft>2pt</PaddingLeft>
      <PaddingRight>2pt</PaddingRight>
      <PaddingTop>2pt</PaddingTop>
      <PaddingBottom>2pt</PaddingBottom>
    </Style>
    <ZIndex>5</ZIndex>
    <CanGrow>true</CanGrow>
    <Left>2.5in</Left>
    <Value>=Count (Fields!ProductID.Value)</Value>
  </Textbox>
  <Textbox Name="textbox5">
    <rd:DefaultName>textbox5</rd:DefaultName>
    <Top>0.125in</Top>
    <Width>1.125in</Width>
    <Style>
      <PaddingLeft>2pt</PaddingLeft>
      <PaddingRight>2pt</PaddingRight>
      <PaddingTop>2pt</PaddingTop>
      <PaddingBottom>2pt</PaddingBottom>
    </Style>
    <ZIndex>4</ZIndex>
    <CanGrow>true</CanGrow>
    <Left>3.125in</Left>
    <Height>0.25in</Height>
    <Value>Total Quantity:</Value>
  </Textbox>
  <Textbox Name="textbox4">
    <rd:DefaultName>textbox4</rd:DefaultName>
    <Top>0.25in</Top>
    <Width>2in</Width>
    <Style>
      <TextAlign>Left</TextAlign>
      <PaddingLeft>2pt</PaddingLeft>
      <PaddingRight>2pt</PaddingRight>
      <PaddingTop>2pt</PaddingTop>
      <PaddingBottom>2pt</PaddingBottom>
    </Style>
    <ZIndex>3</ZIndex>
    <CanGrow>true</CanGrow>

```

```

    <Left>0.5in</Left>
    <Value>Number of Different Products:</Value>
</Textbox>
<Textbox Name="textbox3">
  <rd:DefaultName>textbox3</rd:DefaultName>
  <Width>0.75in</Width>
  <Style>
    <PaddingLeft>2pt</PaddingLeft>
    <PaddingRight>2pt</PaddingRight>
    <PaddingTop>2pt</PaddingTop>
    <PaddingBottom>2pt</PaddingBottom>
  </Style>
  <ZIndex>2</ZIndex>
  <CanGrow>true</CanGrow>
  <Left>0.125in</Left>
  <Height>0.25in</Height>
  <Value>Location:</Value>
</Textbox>
<Textbox Name="textbox2">
  <Top>0.125in</Top>
  <Width>0.875in</Width>
  <Style>
    <BackgroundColor>LightGreen</BackgroundColor>
    <TextAlign>Right</TextAlign>
    <PaddingLeft>2pt</PaddingLeft>
    <PaddingRight>2pt</PaddingRight>
    <PaddingTop>2pt</PaddingTop>
    <PaddingBottom>2pt</PaddingBottom>
  </Style>
  <ZIndex>1</ZIndex>
  <CanGrow>true</CanGrow>
  <Left>4.25in</Left>
  <Height>0.25in</Height>
  <Value>=Sum(Fields!Quantity.Value)</Value>
</Textbox>
<Textbox Name="Name">
  <rd:DefaultName>Name</rd:DefaultName>
  <Width>1.75in</Width>
  <Style>
    <BackgroundColor>LemonChiffon</BackgroundColor>
    <TextAlign>Left</TextAlign>
    <PaddingLeft>2pt</PaddingLeft>
    <PaddingRight>2pt</PaddingRight>
    <PaddingTop>2pt</PaddingTop>
    <PaddingBottom>2pt</PaddingBottom>
  </Style>
  <CanGrow>true</CanGrow>
  <Left>0.875in</Left>
  <Height>0.25in</Height>
  <Value>=Fields!Name.Value</Value>
</Textbox>
</ReportItems>

```

The next elements specify that the **List** is 0.25 inches below the top of its container, which is the report body, and that **List** is 5.25 inches wide. The **Style** element specifies that the entire list is surrounded with a solid border in the default Black color.

```

<Top>0.25in</Top>
<Width>5.25in</Width>
<Style>
  <BorderStyle>
    <Default>Solid</Default>
  </BorderStyle>
</Style>

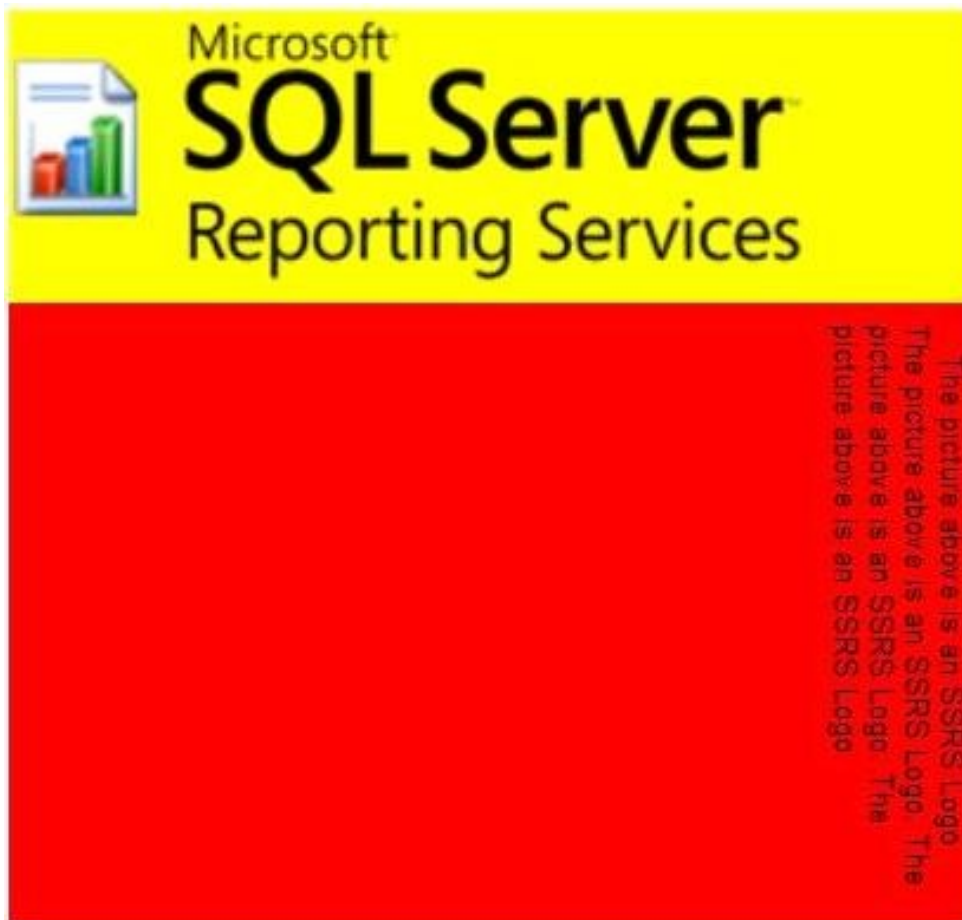
```

The following [Grouping](#) element specifies that the details of the list are grouped by the field named "LocationID". This field is a numerical value that uniquely identifies each location name. Finally, the [Height](#) element specifies that each detail instance of the list is 0.5 inches tall.

```
<Grouping Name="list1_Details_Group">
  <GroupExpressions>
    <GroupExpression>=Fields!LocationID.Value</GroupExpression>
  </GroupExpressions>
</Grouping>
<Height>0.5in</Height>
</List>
```

### 3.16 Rectangle with Image and Textbox

The following figure shows an **image** that was created by using the [Rectangle](#) element and the [Textbox](#) element.



**Figure 24: Rectangle with an image and a text box**

The [Name](#) attribute of the **Rectangle** element is set to "Rectangle1". "Rectangle1" contains two child elements within its [Rectangle.ReportItems](#) collection: an [Image](#) element and a **Textbox** element.

The [Name](#) attribute of the **Image** element within "Rectangle1" is set to "Image2". [Image.Source](#) for "Image2" is set to External, which means that [Image.Value](#) expects an [RdlURL](#) value or an expression that evaluates to the location of an actual image. For "Image2", **Image.Value** is set to

<http://www.widgets.microsoft.com/SSRSLogo.jpg>, which stores the image shown above that shows an icon to the left and the words "Microsoft SQL Server Reporting Services" to the right.

```
<Rectangle Name="Rectangle1">
  <ReportItems>
    <Image Name="Image2">
      <Source>External</Source>
      <Value>http://widgets.microsoft.com/SSRSLogo.jpg</Value>
      <Sizing>FitProportional</Sizing>
      <Height>1.26042in</Height>
      <Width>4.25in</Width>
      <Style>
        <Border>
          <Style>None</Style>
        </Border>
        <PaddingLeft>1pt</PaddingLeft>
        <PaddingRight>1pt</PaddingRight>
        <PaddingTop>1pt</PaddingTop>
        <PaddingBottom>1pt</PaddingBottom>
      </Style>
    </Image>
  </ReportItems>
</Rectangle1>
```

The [Name](#) attribute of the **Textbox** element within "Rectangle1" is set to "Textbox1". "Textbox1" is a simple text box. It contains only one [Paragraph](#) within its [Textbox.Paragraphs](#) collection. Within that one **Paragraph**, "Textbox1" contains only one [TextRun](#) within the [Paragraph.TextRuns](#) collection. Because [Style.WritingMode](#) is set to Vertical for "Textbox1", the text goes from up to down instead of from left to right.

The only **Paragraph** within "Textbox1" has [Paragraph.LeftIndent](#) set to 5pt and [Paragraph.RightIndent](#) set to 5pt. This causes the text within that **Paragraph** to be offset 5 points both to the left and right. However, because **Style.WritingMode** is set to Vertical, the offsets are instead applied to the top and bottom, respectively. Similarly, setting [Paragraph.HangingIndent](#) to 10pt additionally offsets the first line of text within the **Paragraph** 10 points from the top. Finally, because [Textbox.CanGrow](#) is set to true, both "Textbox1" and "Rectangle1" expand vertically beyond their respective heights of 0.65972 inches and 1.94792 inches.

```
<Textbox Name="Textbox1">
  <CanGrow>true</CanGrow>
  <KeepTogether>true</KeepTogether>
  <Paragraphs>
    <Paragraph>
      <TextRuns>
        <TextRun>
          <Value>The picture above is an SSRS Logo. The picture above is an
SSRS
Logo. The picture above is an SSRS Logo. The picture above is an SSRS
Logo.</Value>
          <Style>
            <WritingMode>Vertical</WritingMode>
          </Style>
        </TextRun>
      </TextRuns>
      <LeftIndent>5pt</LeftIndent>
      <RightIndent>5pt</RightIndent>
      <HangingIndent>10pt</HangingIndent>
    </Paragraph>
  </Paragraphs>
  <rd:DefaultName>Textbox1</rd:DefaultName>
  <Top>1.28819in</Top>
  <Height>0.65972in</Height>
  <Width>4.25in</Width>
  <ZIndex>1</ZIndex>
  <Style>
    <WritingMode>Vertical</WritingMode>
  </Style>
</Textbox1>
```

```

    <Border>
      <Style>None</Style>
    </Border>
    <BackgroundColor>Red</BackgroundColor>
    <PaddingLeft>2pt</PaddingLeft>
    <PaddingRight>2pt</PaddingRight>
    <PaddingTop>2pt</PaddingTop>
    <PaddingBottom>2pt</PaddingBottom>
  </Style>
</Textbox>
</ReportItems>
<KeepTogether>true</KeepTogether>
<Top>0.56125in</Top>
<Left>1.11333in</Left>
<Height>1.94792in</Height>
<Width>4.25in</Width>

```

"Rectangle1" has [Style.BackgroundColor](#) set to Yellow. This also applies to "Image2" because the **Image** that is associated with "Image2" contains a transparent background. However, because "Textbox2" has **Style.BackgroundColor** set to Red, the background of "Textbox2" is red instead of yellow, as shown in the preceding figure.

```

<Style>
  <BackgroundColor>Yellow</BackgroundColor>
  <Border>
    <Style>None</Style>
  </Border>
</Style>
</Rectangle>

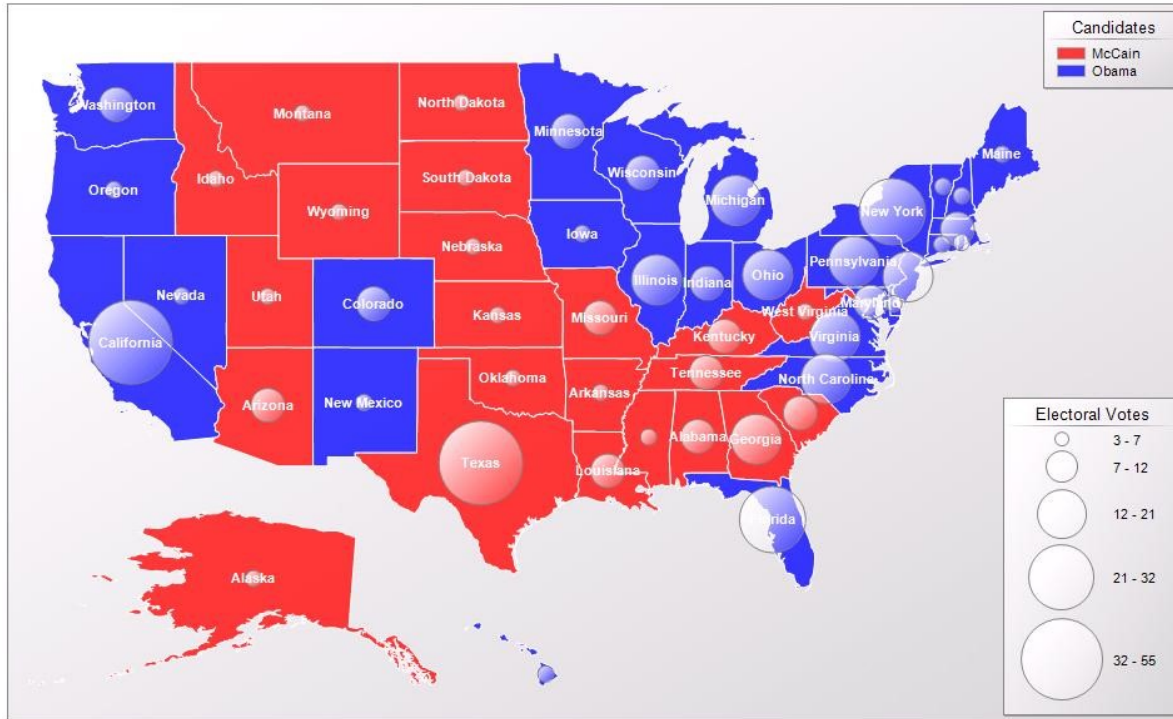
```

### 3.17 Map

The following example report is a **map** that shows the results for the 2008 United States presidential election. The states are colored based on the winning candidate. This map also shows a bubble for each state. The bubble size is based on the number of electoral votes for each state.



## 2008 United States Presidential Election



**Figure 25: Map example**

The data source that is used in this report is "ElectionMap". Two datasets, "Spatial" and "Analytics", are defined in the report. The map report item contains one [MapPolygonLayer](#) that shows the map of the states.

The [MapSpatialDataSet](#) element within the layer describes how the spatial data is retrieved from the "Spatial" dataset. The [MapSpatialDataSet.DataSetName](#) element specifies the name of the dataset to be bound to. The [MapSpatialDataSet.SpatialField](#) element specifies the name of the field to use as the source for the spatial data. The [MapFieldNames](#) element specifies the name of the non-spatial fields to associate with the spatial elements. (Typically, these fields are used to join the spatial elements to the analytical dataset.)

The [MapDataRegion](#) element named "Map1\_PolygonLayer1\_DataRegion" specifies the source for analytical data. In this case, the source is the "Analytics" dataset. The [MapMember](#) element specifies how the rows in the dataset are grouped. In this case, the data is grouped by state by the group expression =Fields!State.Value.

After the spatial data and the analytical data are defined, the [MapBindingFieldPairs](#) element specifies how these data are joined together. Each [MapBindingFieldPair](#) element specifies a FieldName from the spatial data and a BindingExpression from the analytical data. If all the pairs match, the spatial element is bound to the analytical data.

Next is the role of the rules by which to visualize the analytical data. The **polygon layer** contains a [MapPolygonRules](#) element that has a [MapColorRangeRule](#). This rule specifies the range of color to use to colorize the shapes, and a [DataValue](#) element to choose which field to visualize (in this case, =Fields!Winner.Value).

To display the bubble for the number of electoral votes, a [MapPolygonLayer.MapCenterPointRules](#) element is defined in the layer together with the [MapSizeRule](#) element. The **MapSizeRule** element controls the size of the bubbles based on the expression =Sum(Fields!ID\_EV.Value).



```

<?xml version="1.0" encoding="utf-8"?>
<Report xmlns:rd="http://schemas.microsoft.com/SQLServer/reporting/reportdesigner"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2010/01/reportdefinition">
  <DataSources>
    <DataSource Name="ElectionMap">
      <ConnectionProperties>
        <DataProvider>SQL</DataProvider>
        <ConnectionString>Data Source=.\map7;Initial Catalog=Election</ConnectionString>
        <IntegratedSecurity>>true</IntegratedSecurity>
      </ConnectionProperties>
      <rd:DataSourceID>b1c3a34d-af07-4e11-9d13-8df5f2fd0325</rd:DataSourceID>
      <rd:SecurityType>Windows</rd:SecurityType>
    </DataSource>
  </DataSources>
  <DataSets>
    <DataSet Name="Analytics">
      <Fields>
        <Field Name="State">
          <DataField>State</DataField>
          <rd:TypeName>System.String</rd:TypeName>
        </Field>
        <Field Name="ID_EV">
          <DataField>#EV</DataField>
          <rd:TypeName>System.Double</rd:TypeName>
        </Field>
        <Field Name="Date">
          <DataField>Date</DataField>
          <rd:TypeName>System.DateTime</rd:TypeName>
        </Field>
        <Field Name="BO">
          <DataField>BO</DataField>
          <rd:TypeName>System.Double</rd:TypeName>
        </Field>
        <Field Name="JM">
          <DataField>JM</DataField>
          <rd:TypeName>System.Double</rd:TypeName>
        </Field>
        <Field Name="Balance">
          <DataField>Balance</DataField>
          <rd:TypeName>System.Int32</rd:TypeName>
        </Field>
        <Field Name="Winner">
          <DataField>Winner</DataField>
          <rd:TypeName>System.String</rd:TypeName>
        </Field>
      </Fields>
      <Query>
        <DataSourceName>ElectionMap</DataSourceName>
        <CommandText>SELECT
Election.[State]
,Election.#EV
,Election.[Date]
,Election.BO
,Election.JM
,Election.Balance
,Election.Winner
FROM
Election</CommandText>
      <rd:RQDDesignerState>
        <QueryDefinition
xmlns="http://schemas.microsoft.com/ReportingServices/QueryDefinition/Relational">
          <SelectedColumns>
            <ColumnExpression ColumnOwner="Election" ColumnName="State" />
            <ColumnExpression ColumnOwner="Election" ColumnName="#EV" />
            <ColumnExpression ColumnOwner="Election" ColumnName="Date" />
            <ColumnExpression ColumnOwner="Election" ColumnName="BO" />
            <ColumnExpression ColumnOwner="Election" ColumnName="JM" />
            <ColumnExpression ColumnOwner="Election" ColumnName="Balance" />
            <ColumnExpression ColumnOwner="Election" ColumnName="Winner" />
          </SelectedColumns>
        </QueryDefinition>
      </rd:RQDDesignerState>
    </DataSet>
  </DataSets>
</Report>

```

```

        </SelectedColumns>
    </QueryDefinition>
</rd:RQDDesignerState>
</Query>
</DataSet>
<DataSet Name="Spatial">
    <Fields>
        <Field Name="Name">
            <DataField>Name</DataField>
            <rd:TypeName>System.String</rd:TypeName>
        </Field>
        <Field Name="SpatialData">
            <DataField>SpatialData</DataField>
            <rd:TypeName>Microsoft.SqlServer.Types.SqlGeometry,
Microsoft.SqlServer.Types, Version=10.0.0.0, Culture=neutral,
PublicKeyToken=89845dcd8080cc91</rd:TypeName>
        </Field>
    </Fields>
    <Query>
        <DataSourceName>ElectionMap</DataSourceName>
        <CommandText>SELECT
USA.Name
,USA.SpatialData
FROM
USA</CommandText>
    </rd:RQDDesignerState>
    <QueryDefinition
xmlns="http://schemas.microsoft.com/ReportingServices/QueryDefinition/Relational">
        <SelectedColumns>
            <ColumnExpression ColumnOwner="USA" ColumnName="Name" />
            <ColumnExpression ColumnOwner="USA" ColumnName="SpatialData" />
        </SelectedColumns>
    </QueryDefinition>
</rd:RQDDesignerState>
</Query>
</DataSet>
</DataSets>
<ReportSections>
    <ReportSection>
        <Body>
            <ReportItems>
                <Map Name="Map1">
                    <MapViewport>
                        <MapCoordinateSystem>Geographic</MapCoordinateSystem>
                        <MapProjection>Mercator</MapProjection>
                        <ProjectionCenterX>0</ProjectionCenterX>
                        <ProjectionCenterY>0</ProjectionCenterY>
                        <MapLimits>
                            <MinimumX>NaN</MinimumX>
                            <MinimumY>NaN</MinimumY>
                            <MaximumX>NaN</MaximumX>
                            <MaximumY>NaN</MaximumY>
                        </MapLimits>
                        <SimplificationResolution>0.01</SimplificationResolution>
                        <MapCustomView>
                            <CenterX>55.0153007507324</CenterX>
                            <CenterY>50.9827842712402</CenterY>
                            <Zoom>112.46826171875</Zoom>
                        </MapCustomView>
                        <MapMeridians>
                            <Style>
                                <Border>
                                    <Color>#c08080ff</Color>
                                    <Style>None</Style>
                                </Border>
                                <Color>#c08080ff</Color>
                            </Style>
                            <Hidden>true</Hidden>
                            <Interval>NaN</Interval>
                        </MapMeridians>
                    </Map>
                </ReportItems>
            </Body>
        </ReportSection>
    </ReportSections>

```

```

    <ShowLabels>true</ShowLabels>
  </MapMeridians>
  <MapParallels>
    <Style>
      <Border>
        <Color>#c08080ff</Color>
        <Style>None</Style>
      </Border>
      <Color>#c08080ff</Color>
    </Style>
    <Hidden>true</Hidden>
    <Interval>NaN</Interval>
    <ShowLabels>true</ShowLabels>
  </MapParallels>
  <GridUnderContent>false</GridUnderContent>
  <Style>
    <Border>
      <Color>Gray</Color>
      <Style>Solid</Style>
    </Border>
    <BackgroundColor>White</BackgroundColor>
    <BackgroundGradientType>DiagonalLeft</BackgroundGradientType>
    <BackgroundGradientEndColor>
      LightGrey
    </BackgroundGradientEndColor>
    <ShadowOffset>1pt</ShadowOffset>
  </Style>
  <LeftMargin>4pt</LeftMargin>
  <RightMargin>4pt</RightMargin>
  <TopMargin>4pt</TopMargin>
  <BottomMargin>4pt</BottomMargin>
</MapViewport>
<MapDataRegions>
  <MapDataRegion Name="Map1 PolygonLayer1 DataRegion">
    <DataSetName>Analytics</DataSetName>
    <MapMember>
      <Group Name="Map1_PolygonLayer1_Group">
        <GroupExpressions>
          <GroupExpression>=Fields!State.Value</GroupExpression>
        </GroupExpressions>
      </Group>
    </MapMember>
  </MapDataRegion>
</MapDataRegions>
<MapLayers>
  <MapPolygonLayer Name="PolygonLayer1">
    <MapPolygonTemplate>
      <CenterPointOffsetX>0</CenterPointOffsetX>
      <CenterPointOffsetY>0</CenterPointOffsetY>
      <Style>
        <Border>
          <Color>White</Color>
          <Style>Solid</Style>
        </Border>
        <BackgroundColor>White</BackgroundColor>
        <BackgroundGradientType>None</BackgroundGradientType>
        <BackgroundGradientEndColor>
          LightGrey
        </BackgroundGradientEndColor>
        <FontSize>8pt</FontSize>
        <FontWeight>Bold</FontWeight>
        <Color>White</Color>
      </Style>
      <Hidden>false</Hidden>
      <OffsetX>0</OffsetX>
      <OffsetY>0</OffsetY>
      <Label>#NAME</Label>
      <ToolTip />
    </MapPolygonTemplate>
  </MapPolygonLayer>
</MapLayers>
  </DataElementLabel />

```

```

</MapPolygonTemplate>
<MapPolygonRules>
  <MapColorRangeRule>
    <StartColor>#c3ff0000</StartColor>
    <EndColor>#c30000ff</EndColor>
    <ShowInColorScale>>true</ShowInColorScale>
    <DataValue>=Fields!Winner.Value</DataValue>
    <StartValue />
    <EndValue />
    <LegendName>Legend1</LegendName>
    <LegendText>#FROMVALUE{N0} - #TOVALUE{N0}</LegendText>
  </MapColorRangeRule>
</MapPolygonRules>
<MapMarkerTemplate>
  <MapMarker>
    <MapMarkerStyle>Circle</MapMarkerStyle>
  </MapMarker>
  <Style>
    <Border>
      <Color>Gray</Color>
      <Style>Solid</Style>
    </Border>
    <BackgroundColor>White</BackgroundColor>
    <BackgroundGradientType>
      DiagonalLeft
    </BackgroundGradientType>
    <BackgroundGradientEndColor>
      #00ffffff
    </BackgroundGradientEndColor>
  </Style>
  <Hidden>>false</Hidden>
  <OffsetX>0</OffsetX>
  <OffsetY>0</OffsetY>
  <Label />
  <ToolTip />
  <DataElementLabel />
</MapMarkerTemplate>
<MapCenterPointRules>
  <MapSizeRule>
    <StartSize>10pt</StartSize>
    <EndSize>60pt</EndSize>
    <DataValue>=Sum(Fields!ID EV.Value)</DataValue>
    <StartValue />
    <EndValue />
    <LegendName>Legend2</LegendName>
    <LegendText>
      #FROMVALUE - #TOVALUE
    </LegendText>
  </MapSizeRule>
</MapCenterPointRules>
<MapDataRegionName>
Map1 PolygonLayer1 DataRegion
</MapDataRegionName>
<MapBindingFieldPairs>
  <MapBindingFieldPair>
    <FieldName>Name</FieldName>
    <BindingExpression>=Fields!State.Value</BindingExpression>
  </MapBindingFieldPair>
</MapBindingFieldPairs>
<MapFieldDefinitions>
  <MapFieldDefinition>
    <Name>Name</Name>
    <DataType>String</DataType>
  </MapFieldDefinition>
</MapFieldDefinitions>
<MapSpatialDataSet>
  <DataSetName>Spatial</DataSetName>
  <SpatialField>SpatialData</SpatialField>
  <MapFieldNames>

```

```

        <MapFieldName>Name</MapFieldName>
    </MapFieldNames>
</MapSpatialDataSet>
</MapPolygonLayer>
</MapLayers>
<MapLegends>
    <MapLegend Name="Legend1">
        <MapLegendTitle>
            <Style>
                <BackgroundColor>#00ffffff</BackgroundColor>
            </Style>
            <Caption>Candidates</Caption>
        </MapLegendTitle>
        <AutoFitTextDisabled>>false</AutoFitTextDisabled>
        <InterlacedRows>>false</InterlacedRows>
        <EquallySpacedItems>>false</EquallySpacedItems>
        <Position>RightTop</Position>
        <DockOutsideViewport>>false</DockOutsideViewport>
        <Hidden>>false</Hidden>
        <ToolTip />
        <Style>
            <Border>
                <Color>Gray</Color>
                <Style>Solid</Style>
            </Border>
            <BackgroundColor>White</BackgroundColor>
            <BackgroundGradientType>
                DiagonalLeft
            </BackgroundGradientType>
            <BackgroundGradientEndColor>
                LightGrey
            </BackgroundGradientEndColor>
            <ShadowOffset>1pt</ShadowOffset>
        </Style>
        <LeftMargin>4pt</LeftMargin>
        <RightMargin>4pt</RightMargin>
        <TopMargin>4pt</TopMargin>
        <BottomMargin>4pt</BottomMargin>
    </MapLegend>
    <MapLegend Name="Legend2">
        <MapLegendTitle>
            <Style>
                <BackgroundColor>#00ffffff</BackgroundColor>
            </Style>
            <Caption>Electoral Votes</Caption>
        </MapLegendTitle>
        <AutoFitTextDisabled>>false</AutoFitTextDisabled>
        <InterlacedRows>>false</InterlacedRows>
        <EquallySpacedItems>>false</EquallySpacedItems>
        <Position>RightBottom</Position>
        <DockOutsideViewport>>false</DockOutsideViewport>
        <Hidden>>false</Hidden>
        <ToolTip />
        <Style>
            <Border>
                <Color>Gray</Color>
                <Style>Solid</Style>
            </Border>
            <BackgroundColor>White</BackgroundColor>
            <BackgroundGradientType>DiagonalLeft</BackgroundGradientType>
            <BackgroundGradientEndColor>
                LightGrey
            </BackgroundGradientEndColor>
            <ShadowOffset>1pt</ShadowOffset>
        </Style>
        <LeftMargin>4pt</LeftMargin>
        <RightMargin>4pt</RightMargin>
        <TopMargin>4pt</TopMargin>
        <BottomMargin>4pt</BottomMargin>
    </MapLegend>
</MapLegends>

```

```

    </MapLegend>
  </MapLegends>
  <MapDistanceScale>
    <Position>RightBottom</Position>
    <DockOutsideViewport>>false</DockOutsideViewport>
    <Hidden>>true</Hidden>
    <ToolTip />
    <Style>
      <Border>
        <Color>Gray</Color>
        <Style>Solid</Style>
      </Border>
      <BackgroundColor>White</BackgroundColor>
      <BackgroundGradientType>DiagonalLeft</BackgroundGradientType>
      <BackgroundGradientEndColor>
        LightGrey
      </BackgroundGradientEndColor>
      <ShadowOffset>1pt</ShadowOffset>
    </Style>
    <MapSize>
      <Width>80</Width>
      <Height>40</Height>
      <Unit>Point</Unit>
    </MapSize>
    <LeftMargin>4pt</LeftMargin>
    <RightMargin>4pt</RightMargin>
    <TopMargin>4pt</TopMargin>
    <BottomMargin>4pt</BottomMargin>
  </MapDistanceScale>
  <MapColorScale>
    <MapColorScaleTitle>
      <Style />
      <Caption />
    </MapColorScaleTitle>
    <LabelFormat>#,##0.##</LabelFormat>
    <HideEndLabels>>false</HideEndLabels>
    <NoDataText>No data</NoDataText>
    <Position>LeftBottom</Position>
    <DockOutsideViewport>>false</DockOutsideViewport>
    <Hidden>>true</Hidden>
    <ToolTip />
    <Style>
      <Border>
        <Color>Gray</Color>
        <Style>Solid</Style>
      </Border>
      <BackgroundColor>White</BackgroundColor>
      <BackgroundGradientType>DiagonalLeft</BackgroundGradientType>
      <BackgroundGradientEndColor>
        LightGrey
      </BackgroundGradientEndColor>
      <ShadowOffset>1pt</ShadowOffset>
    </Style>
    <LeftMargin>4pt</LeftMargin>
    <RightMargin>4pt</RightMargin>
    <TopMargin>4pt</TopMargin>
    <BottomMargin>4pt</BottomMargin>
  </MapColorScale>
  <MapBorderSkin>
    <Style>
      <Border />
      <BackgroundColor>Gray</BackgroundColor>
      <BackgroundGradientType>None</BackgroundGradientType>
      <BackgroundGradientEndColor>
        White
      </BackgroundGradientEndColor>
      <Color>White</Color>
    </Style>
  </MapBorderSkin>

```

```

<MaximumSpatialElementCount>0</MaximumSpatialElementCount>
<MaximumTotalPointCount>0</MaximumTotalPointCount>
<Top>0.57223in</Top>
<Height>6.09737in</Height>
<Width>9.89583in</Width>
<Style>
  <Border>
    <Color>Gray</Color>
    <Style>None</Style>
  </Border>
  <BackgroundColor>White</BackgroundColor>
</Style>
</Map>
<Textbox Name="ReportTitle">
  <CanGrow>true</CanGrow>
  <KeepTogether>true</KeepTogether>
  <Paragraphs>
    <Paragraph>
      <TextRuns>
        <TextRun>
          <Value>2008 United States Presidential Election</Value>
          <Style>
            <FontStyle>Normal</FontStyle>
            <FontFamily>Verdana</FontFamily>
            <FontSize>16pt</FontSize>
            <FontWeight>Normal</FontWeight>
            <TextDecoration>None</TextDecoration>
          </Style>
        </TextRun>
      </TextRuns>
      <Style />
    </Paragraph>
  </Paragraphs>
  <rd:WatermarkTextbox>Title</rd:WatermarkTextbox>
  <rd:DefaultName>ReportTitle</rd:DefaultName>
  <Top>0.13056in</Top>
  <Left>2.48958in</Left>
  <Height>0.4in</Height>
  <Width>6.66667in</Width>
  <ZIndex>1</ZIndex>
  <Style>
    <Border>
      <Style>None</Style>
    </Border>
    <PaddingLeft>2pt</PaddingLeft>
    <PaddingRight>2pt</PaddingRight>
    <PaddingTop>2pt</PaddingTop>
    <PaddingBottom>2pt</PaddingBottom>
  </Style>
</Textbox>
</ReportItems>
<Height>6.76681in</Height>
<Style>
  <Border>
    <Style>None</Style>
  </Border>
</Style>
</Body>
<Width>10.36333in</Width>
<Page>
  <PageFooter>
    <Height>0.25208in</Height>
    <PrintOnFirstPage>true</PrintOnFirstPage>
    <PrintOnLastPage>true</PrintOnLastPage>
    <Style>
      <Border>
        <Style>None</Style>
      </Border>
    </Style>
  </PageFooter>

```

```

    </PageFooter>
    <LeftMargin>1in</LeftMargin>
    <RightMargin>1in</RightMargin>
    <TopMargin>1in</TopMargin>
    <BottomMargin>1in</BottomMargin>
    <Style />
  </Page>
</ReportSection>
</ReportSections>
<rd:ReportID>bd235c9b-b91c-480f-8759-edc46cbf8e32</rd:ReportID>
<rd:ReportUnitType>Inch</rd:ReportUnitType>
</Report>

```

### 3.18 Matrix

#### **Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The following example of a [Matrix](#) element is an extract from an RDL specification of a report. This example is from a report for the Adventure Works database and shows sales by quarter and product category. This example illustrates the use of a **matrix data region** that provides drilldown from summary data into detail data by showing and hiding rows. The following figures show the matrix in its collapsed state and in a partially expanded state.

The following figure shows the **Matrix** example in its collapsed state. The row headers are the product categories Components, Clothing, Bikes, and Accessories. They are sorted in descending order. The column headers are the years 2002 and 2003; they are sorted in ascending order. The detail cells contain sales data that represent the sum total of all sales for a given category and year. Clicking a plus sign in the category row headers or year column headers expands the row or column to drill down into the data.

	+ 2002	+ 2003
+ Components	\$3,611,041	\$5,489,741
+ Clothing	\$489,820	\$1,024,474
+ Bikes	\$26,664,534	\$35,199,346
+ Accessories	\$93,797	\$595,014

**Figure 26: Example of a matrix with dynamic columns in collapsed state**

The following figure shows the **Matrix** example in a partially expanded state. In the Bikes category, a row header is expanded to show the three subcategories of Mountain Bikes, Road Bikes, and Touring Bikes. Also, the column header 2003 is expanded to show the four quarters of 2003. The cell data now represents the sum total of all sales per subcategory and quarter for those cells that are in a subcategory row and under a quarter column. In a partially expanded matrix, there are also details cells that present data at a higher summarization level. For example, the value \$262,613 is the sum total of all sales for all subcategories of Accessories for the third quarter of 2003. This is because the row Accessories is not expanded.



		2002	2003			
			Q1	Q2	Q3	Q4
Components		\$3,611,041	\$459,086	\$1,111,521	\$2,527,699	\$1,391,434
Clothing		\$489,820	\$106,041	\$192,633	\$395,458	\$330,343
Bikes	Mountain Bikes	\$10,893,468	\$2,517,500	\$2,908,659	\$3,617,012	\$3,808,656
	Road Bikes	\$15,771,066	\$3,584,255	\$4,119,659	\$3,844,124	\$3,734,892
	Touring Bikes				\$3,298,006	\$3,766,585
Accessories		\$93,797	\$15,628	\$32,845	\$262,613	\$283,928

**Figure 27: Example of a matrix with dynamic columns in partially expanded state**

The following is the extract from the RDL report file that produces the matrix that is shown above. The **Matrix** element has its [Name](#) attribute set to "SalesMatrix". It has a single [MatrixColumn](#) in the [MatrixColumns](#) collection that specifies the width of the cells of the detail data to be 0.875 inches. The matrix is bound to the data in the [DataSet](#) "Sales" by the [Matrix.DataSetName](#) element.

```
<Matrix Name="SalesMatrix">
  <MatrixColumns>
    <MatrixColumn>
      <Width>0.875in</Width>
    </MatrixColumn>
  </MatrixColumns>
  <DataSetName>Sales</DataSetName>
  <RowGroupings>
```

This matrix has two [RowGrouping](#) elements in its [RowGroupings](#) collection. This is an ordered list specifying the groupings from outermost to innermost. The first **RowGrouping** element specifies a grouping by category. The width of this **RowGrouping** is set to 1.125 inches, which is the width of the expanded row header cell containing "Bikes" in the preceding figure.

The first **RowGrouping** element has a [DynamicRows](#) element that specifies the content of the row header along with grouping and sorting information. The [Textbox](#) with [Name](#) attribute "Category" has a [Style](#) element that specifies that the row header value is white lettering on a steel blue background. The values that are displayed as row headers come from the [Field](#) named "ProdCat" as is specified in the [Value](#) element of the **Textbox** as the expression =Fields!ProdCat.Value.

There is a [Grouping](#) element with its [Name](#) attribute set to "ProductCategory" that specifies that the data is grouped by the expression "=Fields!ProdCat.Value"; that is, the data is grouped by the "ProdCat" **Field**. Also, there is a [Sorting](#) element that specifies that this group is sorted by the "ProdCat" **Field** in descending order. This is seen in the preceding figure where the categories listed are Components, Clothing, Bikes, and Accessories, which are sorted in reverse alphabetical (descending) order.

```
<RowGrouping>
  <Width>1.125in</Width>
  <DynamicRows>
    <ReportItems>
      <Textbox Name="Category">
        <Style>
          <Color>White</Color>
          <BorderStyle>
            <Default>Solid</Default>
          </BorderStyle>
          <PaddingLeft>2pt</PaddingLeft>
          <PaddingRight>2pt</PaddingRight>
```

```

        <BackgroundColor>SteelBlue</BackgroundColor>
    </Style>
    <ZIndex>2</ZIndex>
    <Value>=Fields!ProdCat.Value</Value>
</Textbox>
</ReportItems>
<Grouping Name="ProductCategory">
    <GroupExpressions>
        <GroupExpression>=Fields!ProdCat.Value</GroupExpression>
    </GroupExpressions>
</Grouping>
<Sorting>
    <SortBy>
        <SortExpression>=Fields!ProdCat.Value</SortExpression>
        <Direction>Descending</Direction>
    </SortBy>
</Sorting>
</DynamicRows>
</RowGrouping>

```

The second **RowGrouping** element specifies a grouping by subcategory. The width of this **RowGrouping** is set to 1.125 inches, which is the width of the row header cell containing Mountain Bikes in the figure. This **RowGrouping** element has a **DynamicRows** element that specifies the content of the row header along with grouping and sorting information. The **Textbox** with **Name** attribute "SubCategory" has a **Style** element that specifies that the row header value is white lettering on a steel blue background. The values to display as row headers come from the **Field** named "SubCat" as is specified in the **Value** element of the **Textbox** as the expression =Fields!SubCat.Value.

This element has something that the previous **RowGrouping** does not have. It has the [Visibility](#) element, which specifies that this row grouping is initially hidden and that the **RowGrouping** that has the **Name** attribute "Category" is used to toggle whether this row grouping is visible or hidden—that is, expanded or collapsed.

The **Grouping** element that has the **Name** attribute "ProductSubcategory" specifies that the data is grouped by the expression =Fields!SubCat.Value, that is, grouped by the **Field** "SubCat". Also, a **Sorting** element specifies that this group is sorted by the **Field** "SubCat" in ascending order. This is shown in the figure where the subcategories are Mountain Bikes, Road Bikes, and Touring Bikes, which are sorted in alphabetical (ascending) order.

```

<RowGrouping>
    <Width>1.125in</Width>
    <DynamicRows>
        <Visibility>
            <Hidden>>true</Hidden>
            <ToggleItem>Category</ToggleItem>
        </Visibility>
        <ReportItems>
            <Textbox Name="SubCategory">
                <Style>
                    <Color>White</Color>
                    <BorderStyle>
                        <Default>Solid</Default>
                    </BorderStyle>
                    <PaddingLeft>2pt</PaddingLeft>
                    <PaddingRight>2pt</PaddingRight>
                    <BackgroundColor>SteelBlue</BackgroundColor>
                </Style>
                <ZIndex>1</ZIndex>
                <Value>=Fields!SubCat.Value</Value>
            </Textbox>
        </ReportItems>
        <Grouping Name="ProductSubcategory">
            <GroupExpressions>
                <GroupExpression>=Fields!SubCat.Value</GroupExpression>
            </GroupExpressions>
        </Grouping>
    </DynamicRows>
</RowGrouping>

```

```

    </GroupExpressions>
  </Grouping>
  <Sorting>
    <SortBy>
      <SortExpression>=Fields!SubCat.Value</SortExpression>
      <Direction>Ascending</Direction>
    </SortBy>
  </Sorting>
</DynamicRows>
</RowGrouping>
</RowGroupings>
<MatrixRows>

```

This matrix has a single [MatrixRow](#) element in its [MatrixRows](#) collection. This **MatrixRow** has a single [MatrixCell](#) element in its [MatrixCells](#) collection. The **Textbox** specifies the content of the details cells of the matrix. It sets the background color to white; the content of the details is an aggregate expression that computes the total sales under the current row headers and column headers of the details cell. That aggregate is specified by the expression =Sum(Fields!Sales.Value).

The height of this **MatrixRow** is set to 0.25 inches, which is the height of a detail cell such as the one containing \$283,928 in the figure.

Additionally, just after the **MatrixRows** element, the **Top** element specifies that the matrix is placed 1.0625 inches below the top of its parent report item.

```

<MatrixRow>
  <MatrixCells>
    <MatrixCell>
      <ReportItems>
        <Textbox Name="Sales">
          <Style>
            <BorderStyle>
              <Default>Solid</Default>
            </BorderStyle>
            <FontSize>9pt</FontSize>
            <Format>C0</Format>
            <PaddingLeft>2pt</PaddingLeft>
            <PaddingRight>2pt</PaddingRight>
            <BackgroundColor>White</BackgroundColor>
          </Style>
          <Value>=Sum(Fields!Sales.Value)</Value>
        </Textbox>
      </ReportItems>
    </MatrixCell>
  </MatrixCells>
  <Height>0.25in</Height>
</MatrixRow>
</MatrixRows>
<Top>1.0625in</Top>
<ColumnGroupings>

```

This matrix has two [ColumnGrouping](#) elements in its [ColumnGroupings](#) collection. This is an ordered list that specifies the groupings from outermost to innermost. The first **ColumnGrouping** element specifies a grouping of sales orders by year. The height of this **ColumnGrouping** is set at the end of this element to 0.25 inches, which is the height of the expanded column header cell containing 2003 in the figure. This element has a [DynamicColumns](#) element that specifies the content of the column header, along with grouping and sorting information. The **Textbox** that has its **Name** attribute set to "OrderYear" has a **Style** element that specifies that the column header value is white lettering on a steel blue background. The values to display as column headers come from the **Field** named "OrderYear" as is specified in the **Value** element of the **Textbox** as the expression =Fields!OrderYear.Value.

A **Grouping** element with its **Name** attribute set to "OrderYear" specifies that the data is grouped by the expression =Fields!OrderYear.Value. That is, the data is grouped by the "OrderYear" **Field**. Also, there is a **Sorting** element that specifies that this group is sorted by the "OrderYear" **Field** in ascending order. This is shown in the preceding figure in which the years that are listed are 2002 and 2003, which are sorted in ascending order.

```
<ColumnGrouping>
  <DynamicColumns>
    <ReportItems>
      <Textbox Name="OrderYear">
        <Style>
          <Color>White</Color>
          <BorderStyle>
            <Default>Solid</Default>
          </BorderStyle>
          <TextAlign>Left</TextAlign>
          <PaddingLeft>2pt</PaddingLeft>
          <PaddingRight>2pt</PaddingRight>
          <BackgroundColor>SteelBlue</BackgroundColor>
        </Style>
        <ZIndex>4</ZIndex>
        <Value>=Fields!OrderYear.Value</Value>
      </Textbox>
    </ReportItems>
    <Grouping Name="OrderYear">
      <GroupExpressions>
        <GroupExpression>=Fields!OrderYear.Value</GroupExpression>
      </GroupExpressions>
    </Grouping>
    <Sorting>
      <SortBy>
        <SortExpression>=Fields!OrderYear.Value</SortExpression>
        <Direction>Ascending</Direction>
      </SortBy>
    </Sorting>
  </DynamicColumns>
  <Height>0.25in</Height>
</ColumnGrouping>
```

The second **ColumnGrouping** element specifies a grouping of the sales orders by quarter. The height of this **ColumnGrouping** is set at the end of this element to 0.25 inches, which is the height of the column header cell containing "Q1" in the preceding figure. This **ColumnGrouping** element has a **DynamicColumns** element that specifies the content of the column header, along with grouping and sorting information. The **Textbox** that has its **Name** attribute set to "OrderQuarter" has a **Style** element that specifies that the column header value is white lettering on a steel-blue background. The values to display as column headers come from the **Field** named "OrderQtr"; this is specified in the **Value** element of the **Textbox** as the expression =Fields! OrderQtr.Value.

This element also has a **Visibility** element, which specifies that this column grouping is initially hidden and that the **ColumnGrouping** that has the **Name** attribute "OrderYear" is used to toggle this column grouping between visible or hidden, that is, expanded or collapsed.

The **Grouping** element that has the **Name** attribute "OrderQtr" specifies that the data is grouped by the expression =Fields!OrderQtr.Value; that is, the data is grouped by the **Field** "OrderQtr". Also, there is a **Sorting** element that specifies that this group is sorted by the **Field** "OrderQtr" in ascending order. This is shown in the figure in which the quarters listed are Q1, Q2, Q3, and Q4, which are sorted in alphabetical (ascending) order.

```
<ColumnGrouping>
  <DynamicColumns>
    <Visibility>
      <Hidden>true</Hidden>
      <ToggleItem>OrderYear</ToggleItem>
    </Visibility>
  </DynamicColumns>
  <Height>0.25in</Height>
</ColumnGrouping>
```

```

</Visibility>
<ReportItems>
  <Textbox Name="OrderQuarter">
    <Style>
      <Color>White</Color>
      <BorderStyle>
        <Default>Solid</Default>
      </BorderStyle>
      <TextAlign>Center</TextAlign>
      <PaddingLeft>2pt</PaddingLeft>
      <PaddingRight>2pt</PaddingRight>
      <BackgroundColor>SteelBlue</BackgroundColor>
    </Style>
    <ZIndex>3</ZIndex>
    <Value>=Fields!OrderQtr.Value</Value>
  </Textbox>
</ReportItems>
<Grouping Name="OrderQtr">
  <GroupExpressions>
    <GroupExpression>=Fields!OrderQtr.Value</GroupExpression>
  </GroupExpressions>
</Grouping>
<Sorting>
  <SortBy>
    <SortExpression>=Fields!OrderQtr.Value</SortExpression>
    <Direction>Ascending</Direction>
  </SortBy>
</Sorting>
</DynamicColumns>
<Height>0.25in</Height>
</ColumnGrouping>
</ColumnGroupings>
<Width>3.125in</Width>
<Corner>

```

This matrix has a [Corner](#) element that consists of a **Textbox** that has no value. Since this **Textbox** does not have a border and has no value, the upper-left corner of the matrix is blank. Just preceding this **Corner** element is the [Width](#) element, which specifies that the initial width of the matrix is 3.125 inches. The actual width of the matrix is derived from the sum of the widths of the component row and column headers.

```

<ReportItems>
  <Textbox Name="textbox1">
    <Style>
      <PaddingLeft>2pt</PaddingLeft>
      <PaddingRight>2pt</PaddingRight>
    </Style>
    <ZIndex>5</ZIndex>
    <Value />
  </Textbox>
</ReportItems>
</Corner>

```

The [Height](#) element specifies that the initial height of the matrix is 0.75 inches. The actual height of the matrix is derived from the sum of the heights of the component row and column headers. The [Left](#) element specifies that the table is placed 0.0625 inches to the right of the left edge of the parent report item that contains this matrix.

```

<Style />
<Height>0.75in</Height>
<Left>0.0625in</Left>
</Matrix>

```

### 3.19 Table

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

The following example of a [Table](#) element is an extract from an RDL specification of a report. This example is from a report for the Adventure Works database and shows year-to-date (YTD) sales and last year's sales for each salesperson grouped by region and sales territory. This example illustrates the use of a **table** data region to provide a tabular grouped listing of data in a grid, providing subtotals for each of the two groups and a grand total for the whole table.

The following figure shows the **Table** example. The first and last rows of the table have a dark blue background color. These are the header and footer rows of the table. The header provides column headers and the footer provides grand totals of YTD sales and last year's sales. The rows with the medium blue background color present subtotals of sales for all the territories within a region. The rows with the light blue background color present the subtotals of sales for all the salespersons within a territory. The rows with the white background color are the detail rows of the table, which provide the sales information for each salesperson.

Region	Sales Territory	Sales Person	Sales YTD	Sales Last Year
Europe	France	Ranjit R Varkey Chudukatil	3827950	2396540
	<i>Subtotal</i>		3827950	2396540
	Germany	Rachel B Valdez	2241204	1307950
	<i>Subtotal</i>		2241204	1307950
	United Kingdom	Jae B Pak	5015682	1635823
	<i>Subtotal</i>		5015682	1635823
<b>Total</b>			<b>11084837</b>	<b>5340313</b>
North America	Canada	José Edvaldo Saraiva	3189356	2038235
		Garrett R Vargas	1764939	1620277
	<i>Subtotal</i>		4954295	3658512
	Central	Jillian Carson	3857164	1997186
	<i>Subtotal</i>		3857164	1997186
	Northeast	Michael G Blythe	4557045	1750406
	<i>Subtotal</i>		4557045	1750406
	Northwest	Pamela O Ansman-Wolfe	0	1927059
		David R Campbell	3587378	1371635
		Tete A Mensa-Annan	1931620	0
	<i>Subtotal</i>		5518999	3298694
	Southeast	Tsvi Michael Reiter	2811013	1849641
	<i>Subtotal</i>		2811013	1849641
	Southwest	Shu K Ito	3018725	2073506
		Linda C Mitchell	5200475	1439156
	<i>Subtotal</i>		8219201	3512662
<b>Total</b>			<b>29917716</b>	<b>16067102</b>
Pacific	Australia	Lynn N Tsoflias	1758386	2278549
	<i>Subtotal</i>		1758386	2278549
<b>Total</b>			<b>1758386</b>	<b>2278549</b>
<b>Grand Total</b>			<b>42760939</b>	<b>23685964</b>

**Figure 28: Example of a table with grouping and totals rows**

The following is the extract from the RDL report file that produces the preceding table. The **Table** element has its **Name** attribute set to "table1". The table is bound to the data in the [DataSet](#) AdventureWorks by the [Table.DataSetName](#) element. The [Top](#) element specifies that the table is 0.33 inches from the top edge of its containing report item, which is the body of the report.

```
<Table Name="table1">
  <DataSetName>AdventureWorks</DataSetName>
  <Top>0.33in</Top>
```

```
<TableGroups>
```

This table has two [TableGroup](#) elements in its [TableGroups](#) collection. This is an ordered list that specifies the groupings from outermost to innermost. The first **TableGroup** element specifies a grouping by region.

The [Grouping](#) element that has its **Name** attribute set to "table1\_Region" that specifies that the data is grouped by the expression =Fields!Region.Value; that is, the data is grouped by the [Field](#) "Region". The [Sorting](#) element specifies that this group is sorted by the **Field** "Region" in ascending order. This is shown in the figure where the regions listed are Europe, North America, and Pacific, which are sorted in alphabetical (ascending) order.

```
<TableGroup>
  <Grouping Name="table1_Region">
    <GroupExpressions>
      <GroupExpression>=Fields!Region.Value</GroupExpression>
    </GroupExpressions>
  </Grouping>
  <Sorting>
    <SortBy>
      <SortExpression>=Fields!Region.Value</SortExpression>
      <Direction>Ascending</Direction>
    </SortBy>
  </Sorting>
```

The [Footer](#) element of the **TableGroup** specifies the rows in the figure that are medium blue in color. These are the lines that represent the total sales for all territories in a region. The [Textbox](#) that has its **Name** attribute value set to "textbox7" specifies the Total label. The text boxes with **Name** attribute values "textbox10" and "textbox11" specify expressions by using the aggregate function [Sum](#) to total all the YTD and last year's sales values in this **grouping scope**.

```
<Footer>
  <TableRows>
    <TableRow>
      <TableCells>
        <TableCell>
          <ReportItems>
            <Textbox Name="textbox7">
              <rd:DefaultName>textbox7</rd:DefaultName>
              <Style>
                <Color>White</Color>
                <BackgroundColor>#60759b</BackgroundColor>
                <BorderColor>
                  <Default>DimGray</Default>
                </BorderColor>
                <BorderStyle>
                  <Default>Solid</Default>
                </BorderStyle>
                <FontFamily>Tahoma</FontFamily>
                <PaddingLeft>2pt</PaddingLeft>
                <PaddingRight>2pt</PaddingRight>
                <PaddingTop>2pt</PaddingTop>
                <PaddingBottom>2pt</PaddingBottom>
              </Style>
              <ZIndex>14</ZIndex>
              <CanGrow>true</CanGrow>
              <Value>Total</Value>
            </Textbox>
          </ReportItems>
        </TableCell>
        <TableCell>
          <ReportItems>
```



```

<Textbox Name="textbox8">
  <rd:DefaultName>textbox8</rd:DefaultName>
  <Style>
    <Color>White</Color>
    <BackgroundColor>#60759b</BackgroundColor>
    <BorderColor>
      <Default>DimGray</Default>
    </BorderColor>
    <BorderStyle>
      <Default>Solid</Default>
    </BorderStyle>
    <FontFamily>Tahoma</FontFamily>
    <PaddingLeft>2pt</PaddingLeft>
    <PaddingRight>2pt</PaddingRight>
    <PaddingTop>2pt</PaddingTop>
    <PaddingBottom>2pt</PaddingBottom>
  </Style>
  <ZIndex>13</ZIndex>
  <CanGrow>true</CanGrow>
  <Value>
  </Value>
</Textbox>
</ReportItems>
</TableCell>
<TableCell>
  <ReportItems>
    <Textbox Name="textbox9">
      <rd:DefaultName>textbox9</rd:DefaultName>
      <Style>
        <Color>White</Color>
        <BackgroundColor>#60759b</BackgroundColor>
        <BorderColor>
          <Default>DimGray</Default>
        </BorderColor>
        <BorderStyle>
          <Default>Solid</Default>
        </BorderStyle>
        <FontFamily>Tahoma</FontFamily>
        <PaddingLeft>2pt</PaddingLeft>
        <PaddingRight>2pt</PaddingRight>
        <PaddingTop>2pt</PaddingTop>
        <PaddingBottom>2pt</PaddingBottom>
      </Style>
      <ZIndex>12</ZIndex>
      <CanGrow>true</CanGrow>
      <Value>
      </Value>
    </Textbox>
  </ReportItems>
</TableCell>
<TableCell>
  <ReportItems>
    <Textbox Name="textbox10">
      <rd:DefaultName>textbox10</rd:DefaultName>
      <Style>
        <Color>White</Color>
        <BackgroundColor>#60759b</BackgroundColor>
        <BorderColor>
          <Default>DimGray</Default>
        </BorderColor>
        <BorderStyle>
          <Default>Solid</Default>
        </BorderStyle>
        <FontFamily>Tahoma</FontFamily>
        <PaddingLeft>2pt</PaddingLeft>
        <PaddingRight>2pt</PaddingRight>
        <PaddingTop>2pt</PaddingTop>
        <PaddingBottom>2pt</PaddingBottom>
      </Style>
    </Textbox>
  </ReportItems>
</TableCell>

```

```

        <ZIndex>11</ZIndex>
        <CanGrow>true</CanGrow>
        <Value>=Round(Sum(Fields!Sales___YTD.Value))</Value>
    </Textbox>
</ReportItems>
</TableCell>
<TableCell>
    <ReportItems>
        <Textbox Name="textbox11">
            <rd:DefaultName>textbox11</rd:DefaultName>
            <Style>
                <Color>White</Color>
                <BackgroundColor>#60759b</BackgroundColor>
                <BorderColor>
                    <Default>DimGray</Default>
                </BorderColor>
                <BorderStyle>
                    <Default>Solid</Default>
                </BorderStyle>
                <FontFamily>Tahoma</FontFamily>
                <PaddingLeft>2pt</PaddingLeft>
                <PaddingRight>2pt</PaddingRight>
                <PaddingTop>2pt</PaddingTop>
                <PaddingBottom>2pt</PaddingBottom>
            </Style>
            <ZIndex>10</ZIndex>
            <CanGrow>true</CanGrow>
            <Value>=Round(Sum(Fields!Sales___Last_Year.Value))</Value>
        </Textbox>
    </ReportItems>
</TableCell>
</TableCells>
<Height>0.21in</Height>
</TableRow>
</TableRows>
</Footer>
</TableGroup>

```

The second **TableGroup** element in the following RDL specifies a grouping by sales territory. There is a **Grouping** element with the **Name** attribute "table1\_Sales\_Territory" that specifies that the data is grouped by the expression =Fields! Sales\_Territory.Value; that is, the data is grouped by the **Field** "Sales\_Territory". A **Sorting** element specifies that this group is sorted by the **Field** Sales\_Territory in ascending order. This is shown in the figure where the sales territories listed are France, Germany, and United Kingdom, under the region Europe, and these are sorted in alphabetical (ascending) order.

```

<TableGroup>
    <Grouping Name="table1_Sales_Territory">
        <GroupExpressions>
            <GroupExpression>=Fields!Sales_Territory.Value</GroupExpression>
        </GroupExpressions>
    </Grouping>
    <Sorting>
        <SortBy>
            <SortExpression>=Fields!Sales_Territory.Value</SortExpression>
            <Direction>Ascending</Direction>
        </SortBy>
    </Sorting>

```

The **Footer** element of this **TableGroup** specifies the rows in the figure that are light blue in color. These are the lines that represent the total sales for all salespersons within a sales territory. The **Textbox** with the **Name** attribute value "textbox13" specifies the label "subtotal". The text boxes with **Name** attribute values "textbox15" and "textbox16" specify expressions by using the aggregate function **Sum** to total all the YTD and last year's sales values in this grouping scope.

```

<Footer>
  <TableRows>
    <TableRow>
      <TableCells>
        <TableCell>
          <ReportItems>
            <Textbox Name="textbox12">
              <rd:DefaultName>textbox12</rd:DefaultName>
              <Style>
                <BackgroundColor>LightSteelBlue</BackgroundColor>
                <BorderColor>
                  <Default>DimGray</Default>
                </BorderColor>
                <BorderStyle>
                  <Default>Solid</Default>
                </BorderStyle>
                <FontFamily>Tahoma</FontFamily>
                <PaddingLeft>2pt</PaddingLeft>
                <PaddingRight>2pt</PaddingRight>
                <PaddingTop>2pt</PaddingTop>
                <PaddingBottom>2pt</PaddingBottom>
              </Style>
              <ZIndex>9</ZIndex>
              <CanGrow>true</CanGrow>
              <Value>
                </Value>
              </Textbox>
            </ReportItems>
          </TableCell>
          <TableCell>
            <ReportItems>
              <Textbox Name="textbox13">
                <rd:DefaultName>textbox13</rd:DefaultName>
                <Style>
                  <BackgroundColor>LightSteelBlue</BackgroundColor>
                  <BorderColor>
                    <Default>DimGray</Default>
                  </BorderColor>
                  <BorderStyle>
                    <Default>Solid</Default>
                  </BorderStyle>
                  <FontStyle>Italic</FontStyle>
                  <FontFamily>Tahoma</FontFamily>
                  <PaddingLeft>2pt</PaddingLeft>
                  <PaddingRight>2pt</PaddingRight>
                  <PaddingTop>2pt</PaddingTop>
                  <PaddingBottom>2pt</PaddingBottom>
                </Style>
                <ZIndex>8</ZIndex>
                <CanGrow>true</CanGrow>
                <Value>Subtotal</Value>
              </Textbox>
            </ReportItems>
          </TableCell>
          <TableCell>
            <ReportItems>
              <Textbox Name="textbox14">
                <rd:DefaultName>textbox14</rd:DefaultName>
                <Style>
                  <BackgroundColor>LightSteelBlue</BackgroundColor>
                  <BorderColor>
                    <Default>DimGray</Default>
                  </BorderColor>
                  <BorderStyle>
                    <Default>Solid</Default>
                  </BorderStyle>
                  <FontFamily>Tahoma</FontFamily>
                  <PaddingLeft>2pt</PaddingLeft>
                  <PaddingRight>2pt</PaddingRight>
                </Style>
              </Textbox>
            </ReportItems>
          </TableCell>
        </TableCells>
      </TableRow>
    </TableRows>
  </Footer>

```

```

        <PaddingTop>2pt</PaddingTop>
        <PaddingBottom>2pt</PaddingBottom>
    </Style>
    <ZIndex>7</ZIndex>
    <CanGrow>>true</CanGrow>
    <Value>
    </Value>
</Textbox>
</ReportItems>
</TableCell>
<TableCell>
    <ReportItems>
        <Textbox Name="textbox15">
            <rd:DefaultName>textbox15</rd:DefaultName>
            <Style>
                <BackgroundColor>LightSteelBlue</BackgroundColor>
                <BorderColor>
                    <Default>DimGray</Default>
                </BorderColor>
                <BorderStyle>
                    <Default>Solid</Default>
                </BorderStyle>
                <FontFamily>Tahoma</FontFamily>
                <PaddingLeft>2pt</PaddingLeft>
                <PaddingRight>2pt</PaddingRight>
                <PaddingTop>2pt</PaddingTop>
                <PaddingBottom>2pt</PaddingBottom>
            </Style>
            <ZIndex>6</ZIndex>
            <CanGrow>>true</CanGrow>
            <Value>=ROUND(Sum(Fields!Sales YTD.Value))</Value>
        </Textbox>
    </ReportItems>
</TableCell>
<TableCell>
    <ReportItems>
        <Textbox Name="textbox16">
            <rd:DefaultName>textbox16</rd:DefaultName>
            <Style>
                <BackgroundColor>LightSteelBlue</BackgroundColor>
                <BorderColor>
                    <Default>DimGray</Default>
                </BorderColor>
                <BorderStyle>
                    <Default>Solid</Default>
                </BorderStyle>
                <FontFamily>Tahoma</FontFamily>
                <PaddingLeft>2pt</PaddingLeft>
                <PaddingRight>2pt</PaddingRight>
                <PaddingTop>2pt</PaddingTop>
                <PaddingBottom>2pt</PaddingBottom>
            </Style>
            <ZIndex>5</ZIndex>
            <CanGrow>>true</CanGrow>

            Value>=Round(Sum(Fields!Sales ___Last_Year.Value))</Value>
        </Textbox>
    </ReportItems>
</TableCell>
</TableCells>
<Height>0.21in</Height>
</TableRow>
</TableRows>
</Footer>
</TableGroup>
</TableGroups>

```

The [Details](#) element of this **Table** specifies the rows in the figure that have a white background color. These are the rows that represent the details of sales for each salesperson within a sales territory. The text boxes with the **Name** attribute values "Region" and "Sales\_Territory" specify the values of the region and sales territory for a given salesperson. The [Textbox.HideDuplicates](#) element specifies that only the first occurrence of each of these will appear within their respective groups. This is why the values North America and Canada are not repeated for the details row for salesperson Garrett R Vargas in the figure. The **Textbox** with the **Name** attribute value "Sales\_Person" specifies the name of the salesperson. The remaining two text boxes specify the salesperson's YTD sales and last year's sales.

```

<Details>
  <TableRows>
    <TableRow>
      <TableCells>
        <TableCell>
          <ReportItems>
            <Textbox Name="Region">
              <rd:DefaultName>Region</rd:DefaultName>
              <HideDuplicates>AdventureWorks</HideDuplicates>
              <Style>
                <BorderColor>
                  <Default>DimGray</Default>
                </BorderColor>
                <BorderStyle>
                  <Default>Solid</Default>
                </BorderStyle>
                <FontFamily>Tahoma</FontFamily>
                <PaddingLeft>2pt</PaddingLeft>
                <PaddingRight>2pt</PaddingRight>
                <PaddingTop>2pt</PaddingTop>
                <PaddingBottom>2pt</PaddingBottom>
              </Style>
              <ZIndex>4</ZIndex>
              <CanGrow>>true</CanGrow>
              <Value>=Fields!Region.Value</Value>
            </Textbox>
          </ReportItems>
        </TableCell>
        <TableCell>
          <ReportItems>
            <Textbox Name="Sales_Territory">
              <rd:DefaultName>Sales_Territory</rd:DefaultName>
              <HideDuplicates>AdventureWorks</HideDuplicates>
              <Style>
                <BorderColor>
                  <Default>DimGray</Default>
                </BorderColor>
                <BorderStyle>
                  <Default>Solid</Default>
                </BorderStyle>
                <FontFamily>Tahoma</FontFamily>
                <PaddingLeft>2pt</PaddingLeft>
                <PaddingRight>2pt</PaddingRight>
                <PaddingTop>2pt</PaddingTop>
                <PaddingBottom>2pt</PaddingBottom>
              </Style>
              <ZIndex>3</ZIndex>
              <CanGrow>>true</CanGrow>
              <Value>=Fields!Sales_Territory.Value</Value>
            </Textbox>
          </ReportItems>
        </TableCell>
        <TableCell>
          <ReportItems>
            <Textbox Name="Sales Person">
              <rd:DefaultName>Sales_Person</rd:DefaultName>
              <Style>

```

```

        <BorderColor>
          <Default>DimGray</Default>
        </BorderColor>
        <BorderStyle>
          <Default>Solid</Default>
        </BorderStyle>
        <FontFamily>Tahoma</FontFamily>
        <PaddingLeft>2pt</PaddingLeft>
        <PaddingRight>2pt</PaddingRight>
        <PaddingTop>2pt</PaddingTop>
        <PaddingBottom>2pt</PaddingBottom>
      </Style>
      <ZIndex>2</ZIndex>
      <CanGrow>true</CanGrow>
      <Value>=Fields!Sales_Person.Value</Value>
    </Textbox>
  </ReportItems>
</TableCell>
<TableCell>
  <ReportItems>
    <Textbox Name="Sales___YTD">
      <rd:DefaultName>Sales___YTD</rd:DefaultName>
      <Style>
        <BorderColor>
          <Default>DimGray</Default>
        </BorderColor>
        <BorderStyle>
          <Default>Solid</Default>
        </BorderStyle>
        <FontFamily>Tahoma</FontFamily>
        <PaddingLeft>2pt</PaddingLeft>
        <PaddingRight>2pt</PaddingRight>
        <PaddingTop>2pt</PaddingTop>
        <PaddingBottom>2pt</PaddingBottom>
      </Style>
      <ZIndex>1</ZIndex>
      <CanGrow>true</CanGrow>
      <Value>=Round(Fields!Sales___YTD.Value)</Value>
    </Textbox>
  </ReportItems>
</TableCell>
<TableCell>
  <ReportItems>
    <Textbox Name="Sales___Last_Year">
      <rd:DefaultName>Sales___Last_Year</rd:DefaultName>
      <Style>
        <BorderColor>
          <Default>DimGray</Default>
        </BorderColor>
        <BorderStyle>
          <Default>Solid</Default>
        </BorderStyle>
        <FontFamily>Tahoma</FontFamily>
        <PaddingLeft>2pt</PaddingLeft>
        <PaddingRight>2pt</PaddingRight>
        <PaddingTop>2pt</PaddingTop>
        <PaddingBottom>2pt</PaddingBottom>
      </Style>
      <CanGrow>true</CanGrow>
      <Value>=Round(Fields!Sales___Last_Year.Value)</Value>
    </Textbox>
  </ReportItems>
</TableCells>
</TableRows>
</Table>

```

The following **Sorting** element specifies that the details are sorted by the **Field** "LastName" in ascending order. We see this in the preceding figure where the salespersons in the Northwest Territory of the North America region are Pamela O Ansmann-Wolfe, David R Campbell, and Tete A Mensa-Annan, who are sorted in alphabetical (ascending) order by their last name. Note that the **Field** "Sales\_Person" is not used to sort the details because this would sort the salespersons by their full name, which would sort them by their first name.

```
<Sorting>
  <SortBy>
    <SortExpression>=Fields!LastName.Value</SortExpression>
    <Direction>Ascending</Direction>
  </SortBy>
</Sorting>
</Details>
```

The **Header** element specifies the header rows of the table, which is the very first row in the figure. This is the dark blue row that contains the column headers.

```
<Header>
  <TableRows>
    <TableRow>
      <TableCells>
        <TableCell>
          <ReportItems>
            <Textbox Name="textbox2">
              <rd:DefaultName>textbox2</rd:DefaultName>
              <Style>
                <Color>White</Color>
                <BackgroundColor>#1c3a70</BackgroundColor>
                <BorderColor>
                  <Default>DimGray</Default>
                </BorderColor>
                <BorderStyle>
                  <Default>Solid</Default>
                </BorderStyle>
                <FontFamily>Tahoma</FontFamily>
                <FontSize>11pt</FontSize>
                <FontWeight>700</FontWeight>
                <PaddingLeft>2pt</PaddingLeft>
                <PaddingRight>2pt</PaddingRight>
                <PaddingTop>2pt</PaddingTop>
                <PaddingBottom>2pt</PaddingBottom>
              </Style>
              <ZIndex>24</ZIndex>
              <CanGrow>true</CanGrow>
              <Value>Region</Value>
            </Textbox>
          </ReportItems>
        </TableCell>
        <TableCell>
          <ReportItems>
            <Textbox Name="textbox3">
              <rd:DefaultName>textbox3</rd:DefaultName>
              <Style>
                <Color>White</Color>
                <BackgroundColor>#1c3a70</BackgroundColor>
                <BorderColor>
                  <Default>DimGray</Default>
                </BorderColor>
                <BorderStyle>
                  <Default>Solid</Default>
                </BorderStyle>
                <FontFamily>Tahoma</FontFamily>
                <FontSize>11pt</FontSize>
                <FontWeight>700</FontWeight>
              </Style>
            </Textbox>
          </ReportItems>
        </TableCell>
      </TableCells>
    </TableRow>
  </TableRows>
</Header>
```

```

        <PaddingLeft>2pt</PaddingLeft>
        <PaddingRight>2pt</PaddingRight>
        <PaddingTop>2pt</PaddingTop>
        <PaddingBottom>2pt</PaddingBottom>
    </Style>
    <ZIndex>23</ZIndex>
    <CanGrow>true</CanGrow>
    <Value>Sales Territory</Value>
</Textbox>
</ReportItems>
</TableCell>
<TableCell>
    <ReportItems>
        <Textbox Name="textbox4">
            <rd:DefaultName>textbox4</rd:DefaultName>
            <Style>
                <Color>White</Color>
                <BackgroundColor>#1c3a70</BackgroundColor>
                <BorderColor>
                    <Default>DimGray</Default>
                </BorderColor>
                <BorderStyle>
                    <Default>Solid</Default>
                </BorderStyle>
                <FontFamily>Tahoma</FontFamily>
                <FontSize>11pt</FontSize>
                <FontWeight>700</FontWeight>
                <PaddingLeft>2pt</PaddingLeft>
                <PaddingRight>2pt</PaddingRight>
                <PaddingTop>2pt</PaddingTop>
                <PaddingBottom>2pt</PaddingBottom>
            </Style>
            <ZIndex>22</ZIndex>
            <CanGrow>true</CanGrow>
            <Value>Sales Person</Value>
        </Textbox>
    </ReportItems>
</TableCell>
<TableCell>
    <ReportItems>
        <Textbox Name="textbox5">
            <rd:DefaultName>textbox5</rd:DefaultName>
            <Style>
                <Color>White</Color>
                <BackgroundColor>#1c3a70</BackgroundColor>
                <BorderColor>
                    <Default>DimGray</Default>
                </BorderColor>
                <BorderStyle>
                    <Default>Solid</Default>
                </BorderStyle>
                <FontFamily>Tahoma</FontFamily>
                <FontSize>11pt</FontSize>
                <FontWeight>700</FontWeight>
                <TextAlign>Right</TextAlign>
                <PaddingLeft>2pt</PaddingLeft>
                <PaddingRight>2pt</PaddingRight>
                <PaddingTop>2pt</PaddingTop>
                <PaddingBottom>2pt</PaddingBottom>
            </Style>
            <ZIndex>21</ZIndex>
            <CanGrow>true</CanGrow>
            <Value>Sales YTD</Value>
        </Textbox>
    </ReportItems>
</TableCell>
<TableCell>
    <ReportItems>
        <Textbox Name="textbox6">

```



```

        <rd:DefaultName>textbox6</rd:DefaultName>
        <Style>
            <Color>White</Color>
            <BackgroundColor>#1c3a70</BackgroundColor>
            <BorderColor>
                <Default>DimGray</Default>
            </BorderColor>
            <BorderStyle>
                <Default>Solid</Default>
            </BorderStyle>
            <FontFamily>Tahoma</FontFamily>
            <FontSize>11pt</FontSize>
            <FontWeight>700</FontWeight>
            <TextAlign>Right</TextAlign>
            <PaddingLeft>2pt</PaddingLeft>
            <PaddingRight>2pt</PaddingRight>
            <PaddingTop>2pt</PaddingTop>
            <PaddingBottom>2pt</PaddingBottom>
        </Style>
        <ZIndex>20</ZIndex>
        <CanGrow>>true</CanGrow>
        <Value>Sales Last Year</Value>
    </Textbox>
</ReportItems>
</TableCell>
</TableCells>
<Height>0.22in</Height>
</TableRow>
</TableRows>
<RepeatOnNewPage>>true</RepeatOnNewPage>
</Header>

```

The [TableColumns](#) element specifies the widths of the five columns in the figure that show the example table.

```

<TableColumns>
  <TableColumn>
    <Width>1.125in</Width>
  </TableColumn>
  <TableColumn>
    <Width>1.125in</Width>
  </TableColumn>
  <TableColumn>
    <Width>1.875in</Width>
  </TableColumn>
  <TableColumn>
    <Width>0.75in</Width>
  </TableColumn>
  <TableColumn>
    <Width>0.75in</Width>
  </TableColumn>
</TableColumns>

```

The **Footer** element specifies the footer rows of the table, which is the very last row in the preceding figure. This is the dark blue row that contains the grand totals. The **Textbox** with the **Name** attribute value "textbox17" specifies the Grand Total label, which appears in the first column of the footer row. The text boxes with **Name** attribute values "textbox20" and "textbox21" specify expressions by using the aggregate function **Sum** to total all the YTD and last year's sales values for all salespersons.

```

<Footer>
  <TableRows>
    <TableRow>
      <TableCells>
        <TableCell>

```

```

<ReportItems>
  <Textbox Name="textbox17">
    <rd:DefaultName>textbox17</rd:DefaultName>
    <Style>
      <Color>White</Color>
      <BackgroundColor>#1c3a70</BackgroundColor>
      <BorderColor>
        <Default>DimGray</Default>
      </BorderColor>
      <BorderStyle>
        <Default>Solid</Default>
      </BorderStyle>
      <FontFamily>Tahoma</FontFamily>
      <FontSize>11pt</FontSize>
      <FontWeight>700</FontWeight>
      <PaddingLeft>2pt</PaddingLeft>
      <PaddingRight>2pt</PaddingRight>
      <PaddingTop>2pt</PaddingTop>
      <PaddingBottom>2pt</PaddingBottom>
    </Style>
    <ZIndex>19</ZIndex>
    <CanGrow>true</CanGrow>
    <Value>Grand Total</Value>
  </Textbox>
</ReportItems>
</TableCell>
<TableCell>
  <ReportItems>
    <Textbox Name="textbox18">
      <rd:DefaultName>textbox18</rd:DefaultName>
      <Style>
        <BackgroundColor>#1c3a70</BackgroundColor>
        <BorderColor>
          <Default>DimGray</Default>
        </BorderColor>
        <BorderStyle>
          <Default>Solid</Default>
        </BorderStyle>
        <PaddingLeft>2pt</PaddingLeft>
        <PaddingRight>2pt</PaddingRight>
        <PaddingTop>2pt</PaddingTop>
        <PaddingBottom>2pt</PaddingBottom>
      </Style>
      <ZIndex>18</ZIndex>
      <CanGrow>true</CanGrow>
      <Value />
    </Textbox>
  </ReportItems>
</TableCell>
<TableCell>
  <ReportItems>
    <Textbox Name="textbox19">
      <rd:DefaultName>textbox19</rd:DefaultName>
      <Style>
        <BackgroundColor>#1c3a70</BackgroundColor>
        <BorderColor>
          <Default>DimGray</Default>
        </BorderColor>
        <BorderStyle>
          <Default>Solid</Default>
        </BorderStyle>
        <PaddingLeft>2pt</PaddingLeft>
        <PaddingRight>2pt</PaddingRight>
        <PaddingTop>2pt</PaddingTop>
        <PaddingBottom>2pt</PaddingBottom>
      </Style>
      <ZIndex>17</ZIndex>
      <CanGrow>true</CanGrow>
      <Value />
    </Textbox>
  </ReportItems>
</TableCell>

```

```

        </Textbox>
    </ReportItems>
</TableCell>
<TableCell>
    <ReportItems>
        <Textbox Name="textbox20">
            <rd:DefaultName>textbox20</rd:DefaultName>
            <Style>
                <Color>White</Color>
                <BackgroundColor>#1c3a70</BackgroundColor>
                <BorderColor>
                    <Default>DimGray</Default>
                </BorderColor>
                <BorderStyle>
                    <Default>Solid</Default>
                </BorderStyle>
                <PaddingLeft>2pt</PaddingLeft>
                <PaddingRight>2pt</PaddingRight>
                <PaddingTop>2pt</PaddingTop>
                <PaddingBottom>2pt</PaddingBottom>
            </Style>
            <ZIndex>16</ZIndex>
            <CanGrow>>true</CanGrow>
            <Value>=ROUND(Sum(Fields!Sales___YTD.Value))</Value>
        </Textbox>
    </ReportItems>
</TableCell>
<TableCell>
    <ReportItems>
        <Textbox Name="textbox21">
            <rd:DefaultName>textbox21</rd:DefaultName>
            <Style>
                <Color>White</Color>
                <BackgroundColor>#1c3a70</BackgroundColor>
                <BorderColor>
                    <Default>DimGray</Default>
                </BorderColor>
                <BorderStyle>
                    <Default>Solid</Default>
                </BorderStyle>
                <PaddingLeft>2pt</PaddingLeft>
                <PaddingRight>2pt</PaddingRight>
                <PaddingTop>2pt</PaddingTop>
                <PaddingBottom>2pt</PaddingBottom>
            </Style>
            <ZIndex>15</ZIndex>
            <CanGrow>>true</CanGrow>
            <Value>=Round(Sum(Fields!Sales    Last Year.Value))</Value>
        </Textbox>
    </ReportItems>
</TableCells>
</TableRows>
</Table>
</Footer>

```

The [Table.Left](#) element specifies that the table is placed 0.125 inches to the right of the left edge of its parent containing report item, which is the report body.

```

    <Left>0.125in</Left>
</Table>

```

## 4 Security

Because Report Definition Language contains only metadata about the report, there are no security considerations for the RDL file itself. Products that implement report processing using RDL as a basis are advised, however, to account for the inherent security risks in the execution of the arbitrary Visual Basic.Net expressions, custom code, and external class references contained within the report.

## 5 Appendix A: RDL XML Schemas

For ease of implementation, the following sections provide the full XML Schemas for this protocol.

Schema name	Prefix	Section
RDL XML Schema for Version 2003/10	xsd	<a href="#">5.1</a>
RDL XML Schema for Version 2005/01	xsd	<a href="#">5.2</a>
RDL XML Schema for Version 2008/01	xsd	<a href="#">5.3</a>
RDL XML Schema for Version 2010/01	xsd	<a href="#">5.4</a>
RDL XML Schema for Version 2011/01*	xsd	<a href="#">5.5</a>
RDL XML Schema for Version 2012/01*	xsd	<a href="#">5.6</a>
RDL XML Schema for Version 2013/01*	xsd	<a href="#">5.7</a>
RDL XML Schema for Version 2016/01	xsd	<a href="#">5.8</a>

\*The micro-versioned RDL XML Schemas for versions 2011/01, 2012/01, and 2013/01 are not standalone schemas and are dependent on the macro-versioned RDL XML Schema for version 2010/01. For more information about macro- and micro-versioned schemas, see section [2.1](#).

### 5.1 RDL XML Schema for Version 2003/10

```
<?xml version="1.0" encoding="utf-8" ?>
<xsd:schema
targetNamespace="http://schemas.microsoft.com/sqlserver/reporting/2003/10/reportdefinition"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2003/10/reportdefinition"
elementFormDefault="qualified">
  <xsd:annotation>
    <xsd:documentation>
```

Microsoft does not make any representation or warranty regarding the schema or any product or item developed based on the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express, implied and statutory warranties, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and freedom from infringement. Without limiting the generality of the foregoing, Microsoft does not make any warranty of any kind that any item developed based on the schema, or any portion of the schema, will not infringe any copyright, patent, trade secret, or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate.

MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES.

(c) Microsoft Corporation. All rights reserved.

```
  </xsd:documentation>
</xsd:annotation>
<xsd:element name="Report">
  <xsd:complexType>
    <xsd:all>
      <xsd:element name="Description" type="xsd:string" minOccurs="0" />
```

```

<xsd:element name="Author" type="xsd:string" minOccurs="0" />
<xsd:element name="AutoRefresh" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="DataSources" type="DataSourcesType" minOccurs="0" />
<xsd:element name="DataSets" type="DataSetsType" minOccurs="0" />
<xsd:element name="Body" type="BodyType" />
<xsd:element name="ReportParameters" type="ReportParametersType"
minOccurs="0" />
<xsd:element name="Custom" type="CustomType" minOccurs="0" />
<xsd:element name="Code" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="SizeType" />
<xsd:element name="PageHeader" type="PageHeaderFooterType"
minOccurs="0" />
<xsd:element name="PageFooter" type="PageHeaderFooterType"
minOccurs="0" />
<xsd:element name="PageHeight" type="SizeType" minOccurs="0" />
<xsd:element name="PageWidth" type="SizeType" minOccurs="0" />
<xsd:element name="LeftMargin" type="SizeType" minOccurs="0" />
<xsd:element name="RightMargin" type="SizeType" minOccurs="0" />
<xsd:element name="TopMargin" type="SizeType" minOccurs="0" />
<xsd:element name="BottomMargin" type="SizeType" minOccurs="0" />
<xsd:element name="EmbeddedImages" type="EmbeddedImagesType"
minOccurs="0" />
<xsd:element name="Language" type="xsd:string" minOccurs="0" />
<xsd:element name="CodeModules" type="CodeModulesType" minOccurs="0" />
<xsd:element name="Classes" type="ClassesType" minOccurs="0" />
<xsd:element name="DataTransform" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataSchema" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementStyle" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="AttributeNormal" />
      <xsd:enumeration value="ElementNormal" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
</xsd:all>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="ReportParametersType">
  <xsd:sequence>
    <xsd:element name="ReportParameter" type="ReportParameterType"
maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ReportParameterType">
  <xsd:all>
    <xsd:element name="DataType">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Boolean" />
          <xsd:enumeration value="DateTime" />
          <xsd:enumeration value="Integer" />
          <xsd:enumeration value="Float" />
          <xsd:enumeration value="String" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Nullable" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DefaultValue" type="DefaultValueType" minOccurs="0" />
    <xsd:element name="AllowBlank" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Prompt" type="xsd:string" minOccurs="0" />
    <xsd:element name="ValidValues" type="ValidValuesType" minOccurs="0" />
    <xsd:element name="MultiValue" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="UsedInQuery" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="False" />
          <xsd:enumeration value="True" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>
</xsd:complexType>

```

```

        <xsd:enumeration value="Auto" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
</xsd:all>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
</xsd:complexType>
<xsd:complexType name="ValidValuesType">
  <xsd:all>
    <xsd:element name="DataSetReference" type="DataSetReferenceType"
      minOccurs="0" />
    <xsd:element name="ParameterValues" type="ParameterValuesType"
      minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="DataSetReferenceType">
  <xsd:all>
    <xsd:element name="DataSetName" type="xsd:string" />
    <xsd:element name="ValueField" type="xsd:string" />
    <xsd:element name="LabelField" type="xsd:string" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="ParameterValuesType">
  <xsd:sequence>
    <xsd:element name="ParameterValue" type="ParameterValueType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ParameterValueType">
  <xsd:all>
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="DefaultValueType">
  <xsd:all>
    <xsd:element name="DataSetReference" type="DataSetReferenceType"
      minOccurs="0" />
    <xsd:element name="Values" type="ValuesType" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="ValuesType">
  <xsd:sequence>
    <xsd:element name="Value" type="xsd:string" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DataSetsType">
  <xsd:sequence>
    <xsd:element name="DataSet" type="DataSetType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DataSetType">
  <xsd:all>
    <xsd:element name="Fields" type="FieldsType" minOccurs="0" />
    <xsd:element name="Query" type="QueryType" />
    <xsd:element name="CaseSensitivity" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="True" />
          <xsd:enumeration value="False" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Collation" type="xsd:string" minOccurs="0" />
    <xsd:element name="AccentSensitivity" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="True" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>

```

```

        <xsd:enumeration value="False" />
        <xsd:enumeration value="Auto" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="KanatypeSensitivity" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="True" />
            <xsd:enumeration value="False" />
            <xsd:enumeration value="Auto" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="WidthSensitivity" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="True" />
            <xsd:enumeration value="False" />
            <xsd:enumeration value="Auto" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
</xsd:all>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
</xsd:complexType>
<xsd:complexType name="FieldsType">
    <xsd:sequence>
        <xsd:element name="Field" type="FieldType" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FieldType">
    <xsd:all>
        <xsd:element name="DataField" type="xsd:string" minOccurs="0" />
        <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    </xsd:all>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
</xsd:complexType>
<xsd:complexType name="QueryType">
    <xsd:all>
        <xsd:element name="DataSourceName" type="xsd:string" />
        <xsd:element name="CommandType" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Text" />
                    <xsd:enumeration value="StoredProcedure" />
                    <xsd:enumeration value="TableDirect" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="CommandText" type="xsd:string" />
        <xsd:element name="QueryParameters" type="QueryParametersType"
            minOccurs="0" />
        <xsd:element name="Timeout" type="xsd:unsignedInt" minOccurs="0" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="DataSourcesType">
    <xsd:sequence>
        <xsd:element name="DataSource" type="DataSourceType"
            maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DataSourceType">
    <xsd:all>
        <xsd:element name="Transaction" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="ConnectionProperties" type="ConnectionPropertiesType"
            minOccurs="0" />
        <xsd:element name="DataSourceReference" type="xsd:string" minOccurs="0" />
    </xsd:all>

```



```

    </xsd:all>
    <xsd:attribute name="Name" type="xsd:string" use="required" />
</xsd:complexType>
<xsd:complexType name="ConnectionPropertiesType">
  <xsd:all>
    <xsd:element name="DataProvider" type="xsd:string" />
    <xsd:element name="ConnectString" type="xsd:string" />
    <xsd:element name="IntegratedSecurity" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Prompt" type="xsd:string" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="QueryParametersType">
  <xsd:sequence>
    <xsd:element name="QueryParameter" type="QueryParameterType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="QueryParameterType">
  <xsd:all>
    <xsd:element name="Value" type="xsd:string" />
  </xsd:all>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
</xsd:complexType>
<xsd:complexType name="CodeModulesType">
  <xsd:sequence>
    <xsd:element name="CodeModule" type="xsd:string" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ClassesType">
  <xsd:sequence>
    <xsd:element name="Class" type="ClassType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ClassType">
  <xsd:all>
    <xsd:element name="ClassName" type="xsd:string" />
    <xsd:element name="InstanceName" type="xsd:normalizedString" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="BodyType">
  <xsd:all>
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="Columns" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="ColumnSpacing" type="SizeType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="PageHeaderFooterType">
  <xsd:all>
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="PrintOnFirstPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PrintOnLastPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="EmbeddedImagesType">
  <xsd:sequence>
    <xsd:element name="EmbeddedImage" type="EmbeddedImageType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="EmbeddedImageType">
  <xsd:all>
    <xsd:element name="MIMEType" type="xsd:string" />
    <xsd:element name="ImageData" type="xsd:string" />
  </xsd:all>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />

```

```

</xsd:complexType>
<xsd:complexType name="ReportItemsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Line" type="LineType" />
    <xsd:element name="Rectangle" type="RectangleType" />
    <xsd:element name="Textbox" type="TextboxType" />
    <xsd:element name="Image" type="ImageType" />
    <xsd:element name="Subreport" type="SubreportType" />
    <xsd:element name="List" type="ListType" />
    <xsd:element name="Matrix" type="MatrixType" />
    <xsd:element name="Table" type="TableType" />
    <xsd:element name="Chart" type="ChartType" />
    <xsd:element name="CustomReportItem" type="CustomReportItemType" />
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="ActionType">
  <xsd:all>
    <xsd:element name="Hyperlink" type="xsd:string" minOccurs="0" />
    <xsd:element name="Drillthrough" type="DrillthroughType" minOccurs="0" />
    <xsd:element name="BookmarkLink" type="xsd:string" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="DrillthroughType">
  <xsd:all>
    <xsd:element name="ReportName" type="xsd:string" />
    <xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
    <xsd:element name="BookmarkLink" type="xsd:string" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="VisibilityType">
  <xsd:all>
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToggleItem" type="xsd:string" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="LineType">
  <xsd:all>
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0"/>
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="Custom" type="CustomType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
          <xsd:enumeration value="Auto"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
</xsd:complexType>
<xsd:complexType name="RectangleType">
  <xsd:all>
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0"/>
  </xsd:all>

```

```

<xsd:element name="Top" type="SizeType" minOccurs="0" />
<xsd:element name="Left" type="SizeType" minOccurs="0" />
<xsd:element name="Height" type="SizeType" minOccurs="0" />
<xsd:element name="Width" type="SizeType" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
<xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
<xsd:element name="Custom" type="CustomType" minOccurs="0" />
<xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output"/>
      <xsd:enumeration value="NoOutput"/>
      <xsd:enumeration value="ContentsOnly"/>
      <xsd:enumeration value="Auto"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
</xsd:all>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
</xsd:complexType>
<xsd:complexType name="TextboxType">
  <xsd:all>
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="Custom" type="CustomType" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="CanGrow" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CanShrink" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="HideDuplicates" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToggleImage" type="ToggleImageType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
          <xsd:enumeration value="Auto"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementStyle" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Auto" />
          <xsd:enumeration value="AttributeNormal" />
          <xsd:enumeration value="ElementNormal" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>

```

```

    </xsd:element>
  </xsd:all>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
</xsd:complexType>
<xsd:complexType name="ToggleImageType">
  <xsd:all>
    <xsd:element name="InitialState" type="xsd:string" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="ImageType">
  <xsd:all>
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="Custom" type="CustomType" minOccurs="0" />
    <xsd:element name="Source">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="External" />
          <xsd:enumeration value="Embedded" />
          <xsd:enumeration value="Database" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="MimeType" type="xsd:string" minOccurs="0" />
    <xsd:element name="Sizing" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="AutoSize" />
          <xsd:enumeration value="Fit" />
          <xsd:enumeration value="FitProportional" />
          <xsd:enumeration value="Clip" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
          <xsd:enumeration value="Auto"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
</xsd:complexType>
<xsd:complexType name="SubreportType">
  <xsd:all>
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0"/>
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />

```

```

<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
<xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0"/>
<xsd:element name="Custom" type="CustomType" minOccurs="0" />
<xsd:element name="ReportName" type="xsd:string" />
<xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
<xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
<xsd:element name="MergeTransactions" type="xsd:boolean" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output"/>
      <xsd:enumeration value="NoOutput"/>
      <xsd:enumeration value="ContentsOnly"/>
      <xsd:enumeration value="Auto"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
</xsd:all>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
</xsd:complexType>
<xsd:complexType name="CustomReportItemType">
  <xsd:all>
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
          <xsd:enumeration value="Auto"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:attribute name="Type" type="xsd:normalizedString" use="required" />
</xsd:complexType>
<xsd:complexType name="ParametersType">
  <xsd:sequence>
    <xsd:element name="Parameter" type="ParameterType"
maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ParameterType">
  <xsd:all>
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="Omit" type="xsd:string" minOccurs="0" />
  </xsd:all>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
</xsd:complexType>
<xsd:complexType name="ListType">
  <xsd:all>
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0"/>
    <xsd:element name="Top" type="SizeType" minOccurs="0" />

```

```

<xsd:element name="Left" type="SizeType" minOccurs="0" />
<xsd:element name="Height" type="SizeType" minOccurs="0" />
<xsd:element name="Width" type="SizeType" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
<xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0"/>
<xsd:element name="Custom" type="CustomType" minOccurs="0" />
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
<xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:element name="Grouping" type="GroupingType" minOccurs="0" />
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
<xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
<xsd:element name="FillPage" type="xsd:boolean" minOccurs="0" />
<xsd:element name="DataInstanceName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataInstanceElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output"/>
      <xsd:enumeration value="NoOutput"/>
      <xsd:enumeration value="ContentsOnly"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output"/>
      <xsd:enumeration value="NoOutput"/>
      <xsd:enumeration value="ContentsOnly"/>
      <xsd:enumeration value="Auto"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
</xsd:all>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
</xsd:complexType>
<xsd:complexType name="GroupingType">
  <xsd:all>
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="GroupExpressions" type="GroupExpressionsType" />
    <xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Custom" type="CustomType" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="Parent" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataCollectionName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
</xsd:complexType>
<xsd:complexType name="GroupExpressionsType">

```

```

<xsd:sequence>
  <xsd:element name="GroupExpression" type="xsd:string"
    maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SortingType">
  <xsd:sequence>
    <xsd:element name="SortBy" type="SortByType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SortByType">
  <xsd:all>
    <xsd:element name="SortExpression" type="xsd:string" />
    <xsd:element name="Direction" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Ascending" />
          <xsd:enumeration value="Descending" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="MatrixType">
  <xsd:all>
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0"/>
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Custom" type="CustomType" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
    <xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="Corner" type="CornerType" minOccurs="0" />
    <xsd:element name="ColumnGroupings" type="ColumnGroupingsType" />
    <xsd:element name="RowGroupings" type="RowGroupingsType" />
    <xsd:element name="MatrixRows" type="MatrixRowsType" />
    <xsd:element name="MatrixColumns" type="MatrixColumnsType" />
    <xsd:element name="LayoutDirection" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="LTR" />
          <xsd:enumeration value="RTL" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GroupsBeforeRowHeaders" type="xsd:unsignedInt"
      minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
          <xsd:enumeration value="Auto"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>
</xsd:complexType>

```

```

    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="CellDataElementName" type="xsd:string" minOccurs="0"/>
  <xsd:element name="CellDataElementOutput" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Output"/>
        <xsd:enumeration value="NoOutput"/>
        <xsd:enumeration value="ContentsOnly"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
</xsd:all>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
</xsd:complexType>
<xsd:complexType name="CornerType">
  <xsd:all>
    <xsd:element name="ReportItems" type="ReportItemsType" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="ColumnGroupingsType">
  <xsd:sequence>
    <xsd:element name="ColumnGrouping" type="ColumnGroupingType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ColumnGroupingType">
  <xsd:all>
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="DynamicColumns" type="DynamicColumnsRowsType"
      minOccurs="0" />
    <xsd:element name="StaticColumns" type="StaticColumnsType"
      minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="DynamicColumnsRowsType">
  <xsd:all>
    <xsd:element name="Grouping" type="GroupingType" />
    <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
    <xsd:element name="Subtotal" type="SubtotalType" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="StaticColumnsType">
  <xsd:sequence>
    <xsd:element name="StaticColumn" type="StaticColumnType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="StaticColumnType">
  <xsd:all>
    <xsd:element name="ReportItems" type="ReportItemsType" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="RowGroupingsType">
  <xsd:sequence>
    <xsd:element name="RowGrouping" type="RowGroupingType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="RowGroupingType">
  <xsd:all>
    <xsd:element name="Width" type="SizeType" />
    <xsd:element name="DynamicRows" type="DynamicColumnsRowsType"
      minOccurs="0" />
    <xsd:element name="StaticRows" type="StaticRowsType" minOccurs="0" />
  </xsd:all>
</xsd:complexType>

```



```

<xsd:complexType name="StaticRowsType">
  <xsd:sequence>
    <xsd:element name="StaticRow" type="StaticRowType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="StaticRowType">
  <xsd:all>
    <xsd:element name="ReportItems" type="ReportItemsType" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="SubtotalType">
  <xsd:all>
    <xsd:element name="ReportItems" type="ReportItemsType" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Position" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Before" />
          <xsd:enumeration value="After" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="MatrixColumnsType">
  <xsd:sequence>
    <xsd:element name="MatrixColumn" type="MatrixColumnType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="MatrixColumnType">
  <xsd:all>
    <xsd:element name="Width" type="SizeType" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="MatrixRowsType">
  <xsd:sequence>
    <xsd:element name="MatrixRow" type="MatrixRowType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="MatrixRowType">
  <xsd:all>
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="MatrixCells" type="MatrixCellsType" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="MatrixCellsType">
  <xsd:sequence>
    <xsd:element name="MatrixCell" type="MatrixCellType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="MatrixCellType">
  <xsd:all>
    <xsd:element name="ReportItems" type="ReportItemsType" />
  </xsd:all>
</xsd:complexType>

```

```

<xsd:complexType name="TableType">
  <xsd:all>
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0"/>
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Custom" type="CustomType" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
    <xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="TableColumns" type="TableColumnsType" />
    <xsd:element name="Header" type="HeaderType" minOccurs="0" />
    <xsd:element name="TableGroups" type="TableGroupsType" minOccurs="0" />
    <xsd:element name="Details" type="DetailsType" minOccurs="0" />
    <xsd:element name="Footer" type="FooterType" minOccurs="0" />
    <xsd:element name="FillPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
          <xsd:enumeration value="Auto"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DetailDataElementName" type="xsd:string"
      minOccurs="0"/>
    <xsd:element name="DetailDataCollectionName" type="xsd:string"
      minOccurs="0"/>
    <xsd:element name="DetailDataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
</xsd:complexType>
<xsd:complexType name="TableColumnsType">
  <xsd:sequence>
    <xsd:element name="TableColumn" type="TableColumnType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="TableColumnType">
  <xsd:all>
    <xsd:element name="Width" type="SizeType" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="HeaderType">
  <xsd:all>

```

```

        <xsd:element name="TableRows" type="TableRowsType" />
        <xsd:element name="RepeatOnNewPage" type="xsd:boolean" minOccurs="0" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="TableRowsType">
    <xsd:sequence>
        <xsd:element name="TableRow" type="TableRowType" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="TableRowType">
    <xsd:all>
        <xsd:element name="TableCells" type="TableCellsType" />
        <xsd:element name="Height" type="SizeType" />
        <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="FooterType">
    <xsd:all>
        <xsd:element name="TableRows" type="TableRowsType" />
        <xsd:element name="RepeatOnNewPage" type="xsd:boolean" minOccurs="0" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="TableGroupsType">
    <xsd:sequence>
        <xsd:element name="TableGroup" type="TableGroupType"
            maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="TableGroupType">
    <xsd:all>
        <xsd:element name="Grouping" type="GroupingType" />
        <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
        <xsd:element name="Header" type="HeaderType" minOccurs="0" />
        <xsd:element name="Footer" type="FooterType" minOccurs="0" />
        <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="DetailsType">
    <xsd:all>
        <xsd:element name="TableRows" type="TableRowsType" />
        <xsd:element name="Grouping" type="GroupingType" minOccurs="0" />
        <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
        <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="TableCellsType">
    <xsd:sequence>
        <xsd:element name="TableCell" type="TableCellType"
            maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="TableCellType">
    <xsd:all>
        <xsd:element name="ReportItems" type="ReportItemsType" />
        <xsd:element name="ColSpan" type="xsd:unsignedInt" minOccurs="0" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="ChartType">
    <xsd:all>
        <xsd:element name="Type" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Column" />
                    <xsd:enumeration value="Bar" />
                    <xsd:enumeration value="Line" />
                    <xsd:enumeration value="Pie" />
                    <xsd:enumeration value="Scatter" />
                    <xsd:enumeration value="Bubble" />
                    <xsd:enumeration value="Area" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:all>
</xsd:complexType>

```

```

        <xsd:enumeration value="Doughnut" />
        <xsd:enumeration value="Stock" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="Subtype" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Stacked" />
            <xsd:enumeration value="PercentStacked" />
            <xsd:enumeration value="Plain" />
            <xsd:enumeration value="Smooth" />
            <xsd:enumeration value="Exploded" />
            <xsd:enumeration value="Line" />
            <xsd:enumeration value="SmoothLine" />
            <xsd:enumeration value="HighLowClose" />
            <xsd:enumeration value="OpenHighLowClose" />
            <xsd:enumeration value="Candlestick" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="Action" type="ActionType" minOccurs="0"/>
<xsd:element name="Top" type="SizeType" minOccurs="0" />
<xsd:element name="Left" type="SizeType" minOccurs="0" />
<xsd:element name="Height" type="SizeType" minOccurs="0" />
<xsd:element name="Width" type="SizeType" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
<xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="Custom" type="CustomType" minOccurs="0" />
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
<xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:element name="SeriesGroupings" type="SeriesGroupingsType"
    minOccurs="0" />
<xsd:element name="CategoryGroupings" type="CategoryGroupingsType"
    minOccurs="0" />
<xsd:element name="ChartData" type="ChartDataTypes" minOccurs="0" />
<xsd:element name="Legend" type="LegendType" minOccurs="0" />
<xsd:element name="CategoryAxis" type="CategoryAxisType" minOccurs="0" />
<xsd:element name="ValueAxis" type="ValueAxisType" minOccurs="0" />
<xsd:element name="Title" type="TitleType" minOccurs="0" />
<xsd:element name="PointWidth" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Palette" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Default"/>
            <xsd:enumeration value="EarthTones"/>
            <xsd:enumeration value="Excel"/>
            <xsd:enumeration value="GrayScale"/>
            <xsd:enumeration value="Light"/>
            <xsd:enumeration value="Pastel"/>
            <xsd:enumeration value="SemiTransparent"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="ThreeDProperties" type="ThreeDPropertiesType"
    minOccurs="0" />
<xsd:element name="PlotArea" type="PlotAreaType" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementOutput" minOccurs="0">
    <xsd:simpleType>

```

```

        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Output"/>
            <xsd:enumeration value="NoOutput"/>
            <xsd:enumeration value="ContentsOnly"/>
            <xsd:enumeration value="Auto"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="ChartElementOutput" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Output"/>
            <xsd:enumeration value="NoOutput"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
</xsd:all>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
</xsd:complexType>
<xsd:complexType name="SeriesGroupingsType">
    <xsd:sequence>
        <xsd:element name="SeriesGrouping" type="SeriesGroupingType"
            maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="SeriesGroupingType">
    <xsd:all>
        <xsd:element name="DynamicSeries" type="DynamicSeriesType"
            minOccurs="0" />
        <xsd:element name="StaticSeries" type="StaticSeriesType" minOccurs="0" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="DynamicSeriesType">
    <xsd:all>
        <xsd:element name="Grouping" type="GroupingType" />
        <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
        <xsd:element name="Label" type="xsd:string" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="StaticSeriesType">
    <xsd:sequence>
        <xsd:element name="StaticMember" type="StaticMemberType"
            maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="StaticMemberType">
    <xsd:all>
        <xsd:element name="Label" type="xsd:string" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="CategoryGroupingsType">
    <xsd:sequence>
        <xsd:element name="CategoryGrouping" type="CategoryGroupingType"
            maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="CategoryGroupingType">
    <xsd:all>
        <xsd:element name="DynamicCategories" type="DynamicCategoriesType"
            minOccurs="0" />
        <xsd:element name="StaticCategories" type="StaticCategoriesType"
            minOccurs="0" />
    </xsd:all>
</xsd:complexType>
<xsd:complexType name="DynamicCategoriesType">
    <xsd:all>
        <xsd:element name="Grouping" type="GroupingType" />
        <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
        <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    </xsd:all>

```

```

    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="StaticCategoriesType">
    <xsd:sequence>
      <xsd:element name="StaticMember" type="StaticMemberType"
        maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="TitleType">
    <xsd:all>
      <xsd:element name="Caption" type="xsd:string" minOccurs="0" />
      <xsd:element name="Style" type="StyleType" minOccurs="0" />
      <xsd:element name="Position" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Center" />
            <xsd:enumeration value="Near" />
            <xsd:enumeration value="Far" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="LegendType">
    <xsd:all>
      <xsd:element name="Visible" type="xsd:boolean" minOccurs="0" />
      <xsd:element name="Style" type="StyleType" minOccurs="0" />
      <xsd:element name="Position" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="TopLeft" />
            <xsd:enumeration value="TopCenter" />
            <xsd:enumeration value="TopRight" />
            <xsd:enumeration value="LeftTop" />
            <xsd:enumeration value="LeftCenter" />
            <xsd:enumeration value="LeftBottom" />
            <xsd:enumeration value="RightTop" />
            <xsd:enumeration value="RightCenter" />
            <xsd:enumeration value="RightBottom" />
            <xsd:enumeration value="BottomLeft" />
            <xsd:enumeration value="BottomCenter" />
            <xsd:enumeration value="BottomRight" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="Layout" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Column" />
            <xsd:enumeration value="Row" />
            <xsd:enumeration value="Table" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="InsidePlotArea" type="xsd:boolean" minOccurs="0" />
    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="CategoryAxisType">
    <xsd:all>
      <xsd:element name="Axis" type="AxisType" minOccurs="0" />
    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="ValueAxisType">
    <xsd:all>
      <xsd:element name="Axis" type="AxisType" minOccurs="0" />
    </xsd:all>
  </xsd:complexType>
  <xsd:complexType name="AxisType">
    <xsd:all>

```

```

<xsd:element name="Visible" type="xsd:boolean" minOccurs="0" />
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="Title" type="TitleType" minOccurs="0" />
<xsd:element name="Margin" type="xsd:boolean" minOccurs="0" />
<xsd:element name="MajorTickMarks" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None" />
      <xsd:enumeration value="Inside" />
      <xsd:enumeration value="Outside" />
      <xsd:enumeration value="Cross" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="MinorTickMarks" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None" />
      <xsd:enumeration value="Inside" />
      <xsd:enumeration value="Outside" />
      <xsd:enumeration value="Cross" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="MajorGridLines" type="MajorGridLinesType"
  minOccurs="0" />
<xsd:element name="MinorGridLines" type="MinorGridLinesType"
  minOccurs="0" />
<xsd:element name="MajorInterval" type="xsd:string" minOccurs="0" />
<xsd:element name="MinorInterval" type="xsd:string" minOccurs="0" />
<xsd:element name="Reverse" type="xsd:boolean" minOccurs="0" />
<xsd:element name="CrossAt" type="xsd:string" minOccurs="0" />
<xsd:element name="Interlaced" type="xsd:boolean" minOccurs="0" />
<xsd:element name="Scalar" type="xsd:boolean" minOccurs="0" />
<xsd:element name="Min" type="xsd:string" minOccurs="0" />
<xsd:element name="Max" type="xsd:string" minOccurs="0" />
<xsd:element name="LogScale" type="xsd:boolean" minOccurs="0" />
</xsd:all>
</xsd:complexType>
<xsd:complexType name="ChartData">
  <xsd:sequence>
    <xsd:element name="ChartSeries" type="ChartSeriesType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="ChartSeriesType">
  <xsd:all>
    <xsd:element name="DataPoints" type="DataPointsType" />
    <xsd:element name="PlotType" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Auto" />
          <xsd:enumeration value="Line" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="DataPointsType">
  <xsd:sequence>
    <xsd:element name="DataPoint" type="DataPointType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DataPointType">
  <xsd:all>
    <xsd:element name="DataValues" type="DataValuesType" />
    <xsd:element name="DataLabel" type="DataLabelType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0" />
  </xsd:all>

```

```

<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="Marker" type="MarkerType" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output"/>
      <xsd:enumeration value="NoOutput"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
</xsd:all>
</xsd:complexType>
<xsd:complexType name="DataValuesType">
  <xsd:sequence>
    <xsd:element name="DataValue" type="DataValueType"
      maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="DataValueType">
  <xsd:all>
    <xsd:element name="Value" type="xsd:string" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="DataLabelType">
  <xsd:all>
    <xsd:element name="Visible" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Auto" />
          <xsd:enumeration value="Top" />
          <xsd:enumeration value="TopLeft" />
          <xsd:enumeration value="TopRight" />
          <xsd:enumeration value="Left" />
          <xsd:enumeration value="Center" />
          <xsd:enumeration value="Right" />
          <xsd:enumeration value="BottomLeft" />
          <xsd:enumeration value="Bottom" />
          <xsd:enumeration value="BottomRight" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Rotation" type="xsd:integer" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="MarkerType">
  <xsd:all>
    <xsd:element name="Type" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Square" />
          <xsd:enumeration value="Circle" />
          <xsd:enumeration value="Diamond" />
          <xsd:enumeration value="Triangle" />
          <xsd:enumeration value="Cross" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Size" type="SizeType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="ThreeDPropertiesType">
  <xsd:all>

```



```

<xsd:element name="Enabled" type="xsd:boolean" minOccurs="0" />
<xsd:element name="ProjectionMode" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Perspective" />
      <xsd:enumeration value="Orthographic" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="Rotation" type="xsd:integer" minOccurs="0" />
<xsd:element name="Inclination" type="xsd:integer" minOccurs="0" />
<xsd:element name="Perspective" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="HeightRatio" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="DepthRatio" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Shading" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None" />
      <xsd:enumeration value="Simple" />
      <xsd:enumeration value="Real" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="GapDepth" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="WallThickness" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="DrawingStyle" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Cube" />
      <xsd:enumeration value="Cylinder" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="Clustered" type="xsd:boolean" minOccurs="0" />
</xsd:all>
</xsd:complexType>
<xsd:complexType name="PlotAreaType">
  <xsd:all>
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="MajorGridLinesType">
  <xsd:all>
    <xsd:element name="ShowGridLines" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="MinorGridLinesType">
  <xsd:all>
    <xsd:element name="ShowGridLines" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="StyleType">
  <xsd:all>
    <xsd:element name="BorderColor" type="BorderColorStyleWidthType"
      minOccurs="0" />
    <xsd:element name="BorderStyle" type="BorderColorStyleWidthType"
      minOccurs="0" />
    <xsd:element name="BorderWidth" type="BorderColorStyleWidthType"
      minOccurs="0" />
    <xsd:element name="BackgroundColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundGradientType" type="xsd:string"
      minOccurs="0" />
    <xsd:element name="BackgroundGradientEndColor" type="xsd:string"
      minOccurs="0" />
    <xsd:element name="BackgroundImage" type="BackgroundImageType"
      minOccurs="0" />
    <xsd:element name="FontStyle" type="xsd:string" minOccurs="0" />
  </xsd:all>

```

```

<xsd:element name="FontFamily" type="xsd:string" minOccurs="0" />
<xsd:element name="FontSize" type="xsd:string" minOccurs="0" />
<xsd:element name="FontWeight" type="xsd:string" minOccurs="0" />
<xsd:element name="Format" type="xsd:string" minOccurs="0" />
<xsd:element name="TextDecoration" type="xsd:string" minOccurs="0" />
<xsd:element name="TextAlign" type="xsd:string" minOccurs="0" />
<xsd:element name="VerticalAlign" type="xsd:string" minOccurs="0" />
<xsd:element name="Color" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingLeft" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingRight" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingTop" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingBottom" type="xsd:string" minOccurs="0" />
<xsd:element name="LineHeight" type="xsd:string" minOccurs="0" />
<xsd:element name="Direction" type="xsd:string" minOccurs="0" />
<xsd:element name="WritingMode" type="xsd:string" minOccurs="0" />
<xsd:element name="Language" type="xsd:string" minOccurs="0" />
<xsd:element name="UnicodeBiDi" type="xsd:string" minOccurs="0" />
<xsd:element name="Calendar" type="xsd:string" minOccurs="0" />
<xsd:element name="NumeralLanguage" type="xsd:string" minOccurs="0" />
<xsd:element name="NumeralVariant" type="xsd:string" minOccurs="0" />
</xsd:all>
</xsd:complexType>
<xsd:complexType name="BorderColorStyleWidthType">
  <xsd:all>
    <xsd:element name="Default" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Right" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Bottom" type="xsd:string" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="BackgroundImageType">
  <xsd:all>
    <xsd:element name="Source">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="External" />
          <xsd:enumeration value="Embedded" />
          <xsd:enumeration value="Database" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundRepeat" type="xsd:string" minOccurs="0" />
  </xsd:all>
</xsd:complexType>
<xsd:complexType name="FiltersType">
  <xsd:sequence>
    <xsd:element name="Filter" type="FilterType" maxOccurs="unbounded" />
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="FilterType">
  <xsd:all>
    <xsd:element name="FilterExpression" type="xsd:string" />
    <xsd:element name="Operator">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Equal" />
          <xsd:enumeration value="Like" />
          <xsd:enumeration value="NotEqual" />
          <xsd:enumeration value="GreaterThan" />
          <xsd:enumeration value="GreaterThanOrEqual" />
          <xsd:enumeration value="LessThan" />
          <xsd:enumeration value="LessThanOrEqual" />
          <xsd:enumeration value="TopN" />
          <xsd:enumeration value="BottomN" />
          <xsd:enumeration value="TopPercent" />
          <xsd:enumeration value="BottomPercent" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:all>

```

```

        <xsd:enumeration value="In" />
        <xsd:enumeration value="Between" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="FilterValues" type="FilterValuesType" />
</xsd:all>
</xsd:complexType>
<xsd:complexType name="FilterValuesType">
    <xsd:sequence>
        <xsd:element name="FilterValue" type="xsd:string" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
<xsd:simpleType name="SizeType">
    <xsd:restriction base="xsd:normalizedString">
        <xsd:minLength value="1" />
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="CustomType">
    <xsd:sequence>
        <xsd:any processContents="skip" minOccurs="0" maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>
</xsd:schema>

```

## 5.2 RDL XML Schema for Version 2005/01

```

<?xml version="1.0" encoding="utf-8" ?>
<xsd:schema
targetNamespace="http://schemas.microsoft.com/sqlserver/reporting/2005/01/reportdefinition"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2005/01/reportdefinition"
elementFormDefault="qualified">
    <xsd:annotation>
        <xsd:documentation>

```

The following schema for Microsoft SQL Server 2005 describes the Report Definition Language (RDL) for defining reports.

Microsoft does not make any representation or warranty regarding the schema or any product or item developed based on the schema. The schema is provided to you on an AS IS basis. Microsoft disclaims all express, implied and statutory warranties, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, and freedom from infringement. Without limiting the generality of the foregoing, Microsoft does not make any warranty of any kind that any item developed based on the schema, or any portion of the schema, will not infringe any copyright, patent, trade secret, or other intellectual property right of any person or entity in any country. It is your responsibility to seek licenses for such intellectual property rights where appropriate.

MICROSOFT SHALL NOT BE LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING WITHOUT LIMITATION, ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING ANY LOST PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES.

(c) Microsoft Corporation. All rights reserved.

```

    </xsd:documentation>
</xsd:annotation>
<xsd:element name="Report">
    <xsd:complexType>
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:element name="Description" type="xsd:string" minOccurs="0" />
            <xsd:element name="Author" type="xsd:string" minOccurs="0" />

```

```

<xsd:element name="AutoRefresh" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="DataSources" type="DataSourcesType" minOccurs="0" />
<xsd:element name="DataSets" type="DataSetsType" minOccurs="0" />
<xsd:element name="Body" type="BodyType" />
<xsd:element name="ReportParameters" type="ReportParametersType"
minOccurs="0" />
<xsd:element name="Code" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="SizeType" />
<xsd:element name="PageHeader" type="PageHeaderFooterType"
minOccurs="0" />
<xsd:element name="PageFooter" type="PageHeaderFooterType"
minOccurs="0" />
<xsd:element name="PageHeight" type="SizeType" minOccurs="0" />
<xsd:element name="PageWidth" type="SizeType" minOccurs="0" />
<xsd:element name="InteractiveHeight" type="SizeType" minOccurs="0" />
<xsd:element name="InteractiveWidth" type="SizeType" minOccurs="0" />
<xsd:element name="LeftMargin" type="SizeType" minOccurs="0" />
<xsd:element name="RightMargin" type="SizeType" minOccurs="0" />
<xsd:element name="TopMargin" type="SizeType" minOccurs="0" />
<xsd:element name="BottomMargin" type="SizeType" minOccurs="0" />
<xsd:element name="EmbeddedImages" type="EmbeddedImagesType"
minOccurs="0" />
<xsd:element name="Language" type="xsd:string" minOccurs="0" />
<xsd:element name="CodeModules" type="CodeModulesType" minOccurs="0" />
<xsd:element name="Classes" type="ClassesType" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType"
minOccurs="0"/>
<xsd:element name="DataTransform" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataSchema" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementStyle" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="AttributeNormal" />
      <xsd:enumeration value="ElementNormal" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
</xsd:element>
<xsd:complexType name="ReportParametersType">
  <xsd:sequence>
    <xsd:element name="ReportParameter" type="ReportParameterType"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ReportParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataType">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Boolean" />
          <xsd:enumeration value="DateTime" />
          <xsd:enumeration value="Integer" />
          <xsd:enumeration value="Float" />
          <xsd:enumeration value="String" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Nullable" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DefaultValue" type="DefaultValueType" minOccurs="0" />
    <xsd:element name="AllowBlank" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Prompt" type="xsd:string" minOccurs="0" />
    <xsd:element name="ValidValues" type="ValidValuesType" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>

```

```

<xsd:element name="MultiValue" type="xsd:boolean" minOccurs="0" />
<xsd:element name="UsedInQuery" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="False" />
      <xsd:enumeration value="True" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ValidValuesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataSetReference" type="DataSetReferenceType"
      minOccurs="0" />
    <xsd:element name="ParameterValues" type="ParameterValuesType"
      minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataSetReferenceType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" />
    <xsd:element name="ValueField" type="xsd:string" />
    <xsd:element name="LabelField" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ParameterValuesType">
  <xsd:sequence>
    <xsd:element name="ParameterValue" type="ParameterValueType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ParameterValueType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DefaultValueType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataSetReference" type="DataSetReferenceType"
      minOccurs="0" />
    <xsd:element name="Values" type="ValuesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ValuesType">
  <xsd:sequence>
    <xsd:element name="Value" type="xsd:string" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataSetsType">
  <xsd:sequence>
    <xsd:element name="DataSet" type="DataSetType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />

```

```

</xsd:complexType>
<xsd:complexType name="DataSetType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Fields" type="FieldsType" minOccurs="0" />
    <xsd:element name="Query" type="QueryType" />
    <xsd:element name="CaseSensitivity" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="True" />
          <xsd:enumeration value="False" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Collation" type="xsd:string" minOccurs="0" />
    <xsd:element name="AccentSensitivity" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="True" />
          <xsd:enumeration value="False" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="KanatypeSensitivity" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="True" />
          <xsd:enumeration value="False" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="WidthSensitivity" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="True" />
          <xsd:enumeration value="False" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="FieldsType">
  <xsd:sequence>
    <xsd:element name="Field" type="FieldType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="FieldType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataField" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="QueryType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataSourceName" type="xsd:string" />
    <xsd:element name="CommandType" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">

```

```

        <xsd:enumeration value="Text" />
        <xsd:enumeration value="StoredProcedure" />
        <xsd:enumeration value="TableDirect" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="CommandText" type="xsd:string" />
<xsd:element name="QueryParameters" type="QueryParametersType"
    minOccurs="0" />
<xsd:element name="Timeout" type="xsd:unsignedInt" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataSourcesType">
    <xsd:sequence>
        <xsd:element name="DataSource" type="DataSourceType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataSourceType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Transaction" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="ConnectionProperties" type="ConnectionPropertiesType"
            minOccurs="0" />
        <xsd:element name="DataSourceReference" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:string" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ConnectionPropertiesType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="DataProvider" type="xsd:string" />
        <xsd:element name="ConnectString" type="xsd:string" />
        <xsd:element name="IntegratedSecurity" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="Prompt" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="QueryParametersType">
    <xsd:sequence>
        <xsd:element name="QueryParameter" type="QueryParameterType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="QueryParameterType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Value" type="xsd:string" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:string" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CodeModulesType">
    <xsd:sequence>
        <xsd:element name="CodeModule" type="xsd:string" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ClassesType">
    <xsd:sequence>
        <xsd:element name="Class" type="ClassType" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

<xsd:complexType name="ClassType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ClassName" type="xsd:string" />
    <xsd:element name="InstanceName" type="xsd:normalizedString" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="BodyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="Columns" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="ColumnSpacing" type="SizeType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="PageHeaderFooterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="PrintOnFirstPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PrintOnLastPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="EmbeddedImagesType">
  <xsd:sequence>
    <xsd:element name="EmbeddedImage" type="EmbeddedImageType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="EmbeddedImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="MIMEType" type="xsd:string" />
    <xsd:element name="ImageData" type="xsd:string" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ReportItemsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Line" type="LineType" />
    <xsd:element name="Rectangle" type="RectangleType" />
    <xsd:element name="Textbox" type="TextboxType" />
    <xsd:element name="Image" type="ImageType" />
    <xsd:element name="Subreport" type="SubreportType" />
    <xsd:element name="List" type="ListType" />
    <xsd:element name="Matrix" type="MatrixType" />
    <xsd:element name="Table" type="TableType" />
    <xsd:element name="Chart" type="ChartType" />
    <xsd:element name="CustomReportItem" type="CustomReportItemType" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ActionType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hyperlink" type="xsd:string" minOccurs="0" />
    <xsd:element name="Drillthrough" type="DrillthroughType" minOccurs="0" />
    <xsd:element name="BookmarkLink" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>

```



```

</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DrillthroughType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportName" type="xsd:string" />
    <xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
    <xsd:element name="BookmarkLink" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="VisibilityType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToggleItem" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="LineType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0"/>
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0"/>
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
          <xsd:enumeration value="Auto"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  <xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="RectangleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0"/>
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

minOccurs="0"/>
<xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output"/>
      <xsd:enumeration value="NoOutput"/>
      <xsd:enumeration value="ContentsOnly"/>
      <xsd:enumeration value="Auto"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TextboxType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0"/>
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="CanGrow" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CanShrink" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="HideDuplicates" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToggleImage" type="ToggleImageType" minOccurs="0" />
    <xsd:element name="UserSort" type="UserSortType" minOccurs="0"/>
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
          <xsd:enumeration value="Auto"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementStyle" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Auto" />
          <xsd:enumeration value="AttributeNormal" />
          <xsd:enumeration value="ElementNormal" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ToggleImageType">

```

```

<xsd:choice minOccurs="1" maxOccurs="unbounded">
  <xsd:element name="InitialState" type="xsd:string" />
  <xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0"/>
  <xsd:element name="Source">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="External" />
        <xsd:enumeration value="Embedded" />
        <xsd:enumeration value="Database" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="Value" type="xsd:string" />
  <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
  <xsd:element name="Sizing" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="AutoSize" />
        <xsd:enumeration value="Fit" />
        <xsd:enumeration value="FitProportional" />
        <xsd:enumeration value="Clip" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
  <xsd:element name="DataElementOutput" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Output"/>
        <xsd:enumeration value="NoOutput"/>
        <xsd:enumeration value="ContentsOnly"/>
        <xsd:enumeration value="Auto"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="SubreportType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0"/>
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
<xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0"/>
<xsd:element name="CustomProperties" type="CustomPropertiesType"
minOccurs="0"/>
<xsd:element name="ReportName" type="xsd:string" />
<xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
<xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
<xsd:element name="MergeTransactions" type="xsd:boolean" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output"/>
      <xsd:enumeration value="NoOutput"/>
      <xsd:enumeration value="ContentsOnly"/>
      <xsd:enumeration value="Auto"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CustomReportItemType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Type" type="xsd:string" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
minOccurs="0"/>
    <xsd:element name="AltReportItem" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="CustomData" type="CustomDataType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
          <xsd:enumeration value="Auto"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CustomDataType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="DataColumnGroupings" type="DataColumnGroupingsType"
minOccurs="0" />
    <xsd:element name="DataRowGroupings" type="DataRowGroupingsType"
minOccurs="0" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

        <xsd:element name="DataRows" type="DataRowsType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataColumnGroupingsType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="DataGroupings" type="DataGroupingsType" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataRowGroupingsType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="DataGroupings" type="DataGroupingsType" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataGroupingsType">
    <xsd:sequence>
        <xsd:element name="DataGrouping" type="DataGroupingType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataGroupingType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Static" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="Grouping" type="GroupingType" minOccurs="0" />
        <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
        <xsd:element name="Subtotal" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType"
            minOccurs="0"/>
        <xsd:element name="DataGroupings" type="DataGroupingsType"
            minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataRowsType">
    <xsd:sequence>
        <xsd:element name="DataRow" type="DataRowType" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataRowType">
    <xsd:sequence>
        <xsd:element name="DataCell" type="DataCellType" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataCellType">
    <xsd:sequence>
        <xsd:element name="DataValue" type="DataValueType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ParametersType">
    <xsd:sequence>
        <xsd:element name="Parameter" type="ParameterType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ParameterType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">

```

```

    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="Omit" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ListType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0"/>
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0"/>
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0"/>
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
    <xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="Grouping" type="GroupingType" minOccurs="0" />
    <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="FillPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DataInstanceName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataInstanceElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
          <xsd:enumeration value="ContentsOnly"/>
          <xsd:enumeration value="Auto"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="GroupingType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="GroupExpressions" type="GroupExpressionsType" />
    <xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0"/>
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="Parent" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataCollectionName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output"/>
      <xsd:enumeration value="NoOutput"/>
      <xsd:enumeration value="ContentsOnly"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="GroupExpressionsType">
  <xsd:sequence>
    <xsd:element name="GroupExpression" type="xsd:string"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="SortingType">
  <xsd:sequence>
    <xsd:element name="SortBy" type="SortByType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="SortByType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="SortExpression" type="xsd:string" />
    <xsd:element name="Direction" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Ascending" />
          <xsd:enumeration value="Descending" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="MatrixType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0"/>
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0"/>
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0"/>
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
    <xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="Corner" type="CornerType" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="ColumnGroupings" type="ColumnGroupingsType" />
<xsd:element name="RowGroupings" type="RowGroupingsType" />
<xsd:element name="MatrixRows" type="MatrixRowsType" />
<xsd:element name="MatrixColumns" type="MatrixColumnsType" />
<xsd:element name="LayoutDirection" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="LTR" />
      <xsd:enumeration value="RTL" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="GroupsBeforeRowHeaders" type="xsd:unsignedInt"
  minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output"/>
      <xsd:enumeration value="NoOutput"/>
      <xsd:enumeration value="ContentsOnly"/>
      <xsd:enumeration value="Auto"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="CellDataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="CellDataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output"/>
      <xsd:enumeration value="NoOutput"/>
      <xsd:enumeration value="ContentsOnly"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CornerType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ColumnGroupingsType">
  <xsd:sequence>
    <xsd:element name="ColumnGrouping" type="ColumnGroupingType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ColumnGroupingType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="FixedHeader" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DynamicColumns" type="DynamicColumnsRowsType"
      minOccurs="0" />
    <xsd:element name="StaticColumns" type="StaticColumnsType"
      minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DynamicColumnsRowsType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Grouping" type="GroupingType" />
  </xsd:choice>

```



```

    <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
    <xsd:element name="Subtotal" type="SubtotalType" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="StaticColumnsType">
  <xsd:sequence>
    <xsd:element name="StaticColumn" type="StaticColumnType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="StaticColumnType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="RowGroupingsType">
  <xsd:sequence>
    <xsd:element name="RowGrouping" type="RowGroupingType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="RowGroupingType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Width" type="SizeType" />
    <xsd:element name="FixedHeader" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DynamicRows" type="DynamicColumnsRowsType"
      minOccurs="0" />
    <xsd:element name="StaticRows" type="StaticRowsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="StaticRowsType">
  <xsd:sequence>
    <xsd:element name="StaticRow" type="StaticRowType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="StaticRowType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="SubtotalType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Position" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Before" />
          <xsd:enumeration value="After" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>

```

```

        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Output"/>
            <xsd:enumeration value="NoOutput"/>
            <xsd:enumeration value="ContentsOnly"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="MatrixColumnsType">
    <xsd:sequence>
        <xsd:element name="MatrixColumn" type="MatrixColumnType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="MatrixColumnType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Width" type="SizeType" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="MatrixRowsType">
    <xsd:sequence>
        <xsd:element name="MatrixRow" type="MatrixRowType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="MatrixRowType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Height" type="SizeType" />
        <xsd:element name="MatrixCells" type="MatrixCellsType" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="MatrixCellsType">
    <xsd:sequence>
        <xsd:element name="MatrixCell" type="MatrixCellType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="MatrixCellType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="ReportItems" type="ReportItemsType" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TableType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="Action" type="ActionType" minOccurs="0"/>
        <xsd:element name="Top" type="SizeType" minOccurs="0" />
        <xsd:element name="Left" type="SizeType" minOccurs="0" />
        <xsd:element name="Height" type="SizeType" minOccurs="0" />
        <xsd:element name="Width" type="SizeType" minOccurs="0" />
        <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
        <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
        <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
        <xsd:element name="Label" type="xsd:string" minOccurs="0" />
        <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
        <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
        <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0"/>
    </xsd:choice>

```

```

<xsd:element name="CustomProperties" type="CustomPropertiesType"
  minOccurs="0"/>
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
<xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:element name="TableColumns" type="TableColumnsType" />
<xsd:element name="Header" type="HeaderType" minOccurs="0" />
<xsd:element name="TableGroups" type="TableGroupsType" minOccurs="0" />
<xsd:element name="Details" type="DetailsType" minOccurs="0" />
<xsd:element name="Footer" type="FooterType" minOccurs="0" />
<xsd:element name="FillPage" type="xsd:boolean" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output"/>
      <xsd:enumeration value="NoOutput"/>
      <xsd:enumeration value="ContentsOnly"/>
      <xsd:enumeration value="Auto"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="DetailDataElementName" type="xsd:string"
  minOccurs="0"/>
<xsd:element name="DetailDataCollectionName" type="xsd:string"
  minOccurs="0"/>
<xsd:element name="DetailDataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output"/>
      <xsd:enumeration value="NoOutput"/>
      <xsd:enumeration value="ContentsOnly"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TableColumnsType">
  <xsd:sequence>
    <xsd:element name="TableColumn" type="TableColumnType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TableColumnType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Width" type="SizeType" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="FixedHeader" type="xsd:boolean" minOccurs="0" />
  <xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="HeaderType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TableRows" type="TableRowsType" />
    <xsd:element name="FixedHeader" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="RepeatOnNewPage" type="xsd:boolean" minOccurs="0" />
  <xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TableRowsType">

```

```

<xsd:sequence>
  <xsd:element name="TableRow" type="TableRowType" maxOccurs="unbounded" />
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TableRowType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TableCells" type="TableCellsType" />
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="FooterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TableRows" type="TableRowsType" />
    <xsd:element name="RepeatOnNewPage" type="xsd:boolean" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TableGroupsType">
  <xsd:sequence>
    <xsd:element name="TableGroup" type="TableGroupType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TableGroupType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Grouping" type="GroupingType" />
    <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
    <xsd:element name="Header" type="HeaderType" minOccurs="0" />
    <xsd:element name="Footer" type="FooterType" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DetailsType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TableRows" type="TableRowsType" />
    <xsd:element name="Grouping" type="GroupingType" minOccurs="0" />
    <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TableCellsType">
  <xsd:sequence>
    <xsd:element name="TableCell" type="TableCellType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TableCellType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" />
    <xsd:element name="ColSpan" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>

```

```

    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Column" />
      <xsd:enumeration value="Bar" />
      <xsd:enumeration value="Line" />
      <xsd:enumeration value="Pie" />
      <xsd:enumeration value="Scatter" />
      <xsd:enumeration value="Bubble" />
      <xsd:enumeration value="Area" />
      <xsd:enumeration value="Doughnut" />
      <xsd:enumeration value="Stock" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="Subtype" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Stacked" />
      <xsd:enumeration value="PercentStacked" />
      <xsd:enumeration value="Plain" />
      <xsd:enumeration value="Smooth" />
      <xsd:enumeration value="Exploded" />
      <xsd:enumeration value="Line" />
      <xsd:enumeration value="SmoothLine" />
      <xsd:enumeration value="HighLowClose" />
      <xsd:enumeration value="OpenHighLowClose" />
      <xsd:enumeration value="Candlestick" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="Action" type="ActionType" minOccurs="0"/>
<xsd:element name="Top" type="SizeType" minOccurs="0" />
<xsd:element name="Left" type="SizeType" minOccurs="0" />
<xsd:element name="Height" type="SizeType" minOccurs="0" />
<xsd:element name="Width" type="SizeType" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
<xsd:element name="LinkToChild" type="xsd:string" minOccurs="0"/>
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType"
  minOccurs="0"/>
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
<xsd:element name="NoRows" type="xsd:string" minOccurs="0" />
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
<xsd:element name="PageBreakAtStart" type="xsd:boolean" minOccurs="0" />
<xsd:element name="PageBreakAtEnd" type="xsd:boolean" minOccurs="0" />
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:element name="SeriesGroupings" type="SeriesGroupingsType"
  minOccurs="0" />
<xsd:element name="CategoryGroupings" type="CategoryGroupingsType"
  minOccurs="0" />
<xsd:element name="ChartData" type="ChartDataTypes" minOccurs="0" />
<xsd:element name="Legend" type="LegendType" minOccurs="0" />
<xsd:element name="CategoryAxis" type="CategoryAxisType" minOccurs="0" />
<xsd:element name="ValueAxis" type="ValueAxisType" minOccurs="0" />
<xsd:element name="Title" type="TitleType" minOccurs="0" />
<xsd:element name="PointWidth" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Palette" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Default"/>
      <xsd:enumeration value="EarthTones"/>
      <xsd:enumeration value="Excel"/>
      <xsd:enumeration value="GrayScale"/>
      <xsd:enumeration value="Light"/>
      <xsd:enumeration value="Pastel"/>
      <xsd:enumeration value="SemiTransparent"/>
    </xsd:restriction>
  </xsd:simpleType>

```

```

        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="ThreeDProperties" type="ThreeDPropertiesType"
    minOccurs="0" />
<xsd:element name="PlotArea" type="PlotAreaType" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
<xsd:element name="DataElementOutput" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Output"/>
            <xsd:enumeration value="NoOutput"/>
            <xsd:enumeration value="ContentsOnly"/>
            <xsd:enumeration value="Auto"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="ChartElementOutput" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Output"/>
            <xsd:enumeration value="NoOutput"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="SeriesGroupingsType">
    <xsd:sequence>
        <xsd:element name="SeriesGrouping" type="SeriesGroupingType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="SeriesGroupingType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="DynamicSeries" type="DynamicSeriesType"
            minOccurs="0" />
        <xsd:element name="StaticSeries" type="StaticSeriesType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DynamicSeriesType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Grouping" type="GroupingType" />
        <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
        <xsd:element name="Label" type="xsd:string" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="StaticSeriesType">
    <xsd:sequence>
        <xsd:element name="StaticMember" type="StaticMemberType"
            maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="StaticMemberType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Label" type="xsd:string" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

<xsd:complexType name="CategoryGroupingsType">
  <xsd:sequence>
    <xsd:element name="CategoryGrouping" type="CategoryGroupingType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CategoryGroupingType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DynamicCategories" type="DynamicCategoriesType"
      minOccurs="0" />
    <xsd:element name="StaticCategories" type="StaticCategoriesType"
      minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DynamicCategoriesType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Grouping" type="GroupingType" />
    <xsd:element name="Sorting" type="SortingType" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="StaticCategoriesType">
  <xsd:sequence>
    <xsd:element name="StaticMember" type="StaticMemberType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TitleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Caption" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Position" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Center" />
          <xsd:enumeration value="Near" />
          <xsd:enumeration value="Far" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="LegendType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Visible" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Position" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="TopLeft" />
          <xsd:enumeration value="TopCenter" />
          <xsd:enumeration value="TopRight" />
          <xsd:enumeration value="LeftTop" />
          <xsd:enumeration value="LeftCenter" />
          <xsd:enumeration value="LeftBottom" />
          <xsd:enumeration value="RightTop" />
          <xsd:enumeration value="RightCenter" />
          <xsd:enumeration value="RightBottom" />
          <xsd:enumeration value="BottomLeft" />
          <xsd:enumeration value="BottomCenter" />
          <xsd:enumeration value="BottomRight" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="Layout" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Column" />
            <xsd:enumeration value="Row" />
            <xsd:enumeration value="Table" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="InsidePlotArea" type="xsd:boolean" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CategoryAxisType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Axis" type="AxisType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ValueAxisType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Axis" type="AxisType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="AxisType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Visible" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="Title" type="TitleType" minOccurs="0" />
        <xsd:element name="Margin" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="MajorTickMarks" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="None" />
                    <xsd:enumeration value="Inside" />
                    <xsd:enumeration value="Outside" />
                    <xsd:enumeration value="Cross" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="MinorTickMarks" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="None" />
                    <xsd:enumeration value="Inside" />
                    <xsd:enumeration value="Outside" />
                    <xsd:enumeration value="Cross" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="MajorGridLines" type="MajorGridLinesType"
            minOccurs="0" />
        <xsd:element name="MinorGridLines" type="MinorGridLinesType"
            minOccurs="0" />
        <xsd:element name="MajorInterval" type="xsd:string" minOccurs="0" />
        <xsd:element name="MinorInterval" type="xsd:string" minOccurs="0" />
        <xsd:element name="Reverse" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="CrossAt" type="xsd:string" minOccurs="0" />
        <xsd:element name="Interlaced" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="Scalar" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="Min" type="xsd:string" minOccurs="0" />
        <xsd:element name="Max" type="xsd:string" minOccurs="0" />
    </xsd:choice>

```



```

    <xsd:element name="LogScale" type="xsd:boolean" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartData">
  <xsd:sequence>
    <xsd:element name="ChartSeries" type="ChartSeriesType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartSeriesType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataPoints" type="DataPointsType" />
    <xsd:element name="PlotType" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Auto" />
          <xsd:enumeration value="Line" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataPointsType">
  <xsd:sequence>
    <xsd:element name="DataPoint" type="DataPointType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataPointType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataValues" type="DataValuesType" />
    <xsd:element name="DataLabel" type="DataLabelType" minOccurs="0" />
    <xsd:element name="Action" type="ActionType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Marker" type="MarkerType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"/>
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output"/>
          <xsd:enumeration value="NoOutput"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
  <xsd:any namespace="##other" processContents="skip"/>
</xsd:complexType>
<xsd:complexType name="DataValuesType">
  <xsd:sequence>
    <xsd:element name="DataValue" type="DataValueType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataValueType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Value" type="xsd:string" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

<xsd:complexType name="DataLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Visible" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Auto" />
          <xsd:enumeration value="Top" />
          <xsd:enumeration value="TopLeft" />
          <xsd:enumeration value="TopRight" />
          <xsd:enumeration value="Left" />
          <xsd:enumeration value="Center" />
          <xsd:enumeration value="Right" />
          <xsd:enumeration value="BottomLeft" />
          <xsd:enumeration value="Bottom" />
          <xsd:enumeration value="BottomRight" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Rotation" type="xsd:integer" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="MarkerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Type" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Square" />
          <xsd:enumeration value="Circle" />
          <xsd:enumeration value="Diamond" />
          <xsd:enumeration value="Triangle" />
          <xsd:enumeration value="Cross" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Size" type="SizeType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ThreeDPropertiesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Enabled" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ProjectionMode" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Perspective" />
          <xsd:enumeration value="Orthographic" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Rotation" type="xsd:integer" minOccurs="0" />
    <xsd:element name="Inclination" type="xsd:integer" minOccurs="0" />
    <xsd:element name="Perspective" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="HeightRatio" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="DepthRatio" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Shading" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Simple" />
          <xsd:enumeration value="Real" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="GapDepth" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="WallThickness" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="DrawingStyle" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Cube" />
            <xsd:enumeration value="Cylinder" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="Clustered" type="xsd:boolean" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="PlotAreaType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="MajorGridLinesType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="ShowGridLines" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="MinorGridLinesType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="ShowGridLines" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="StyleType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="BorderColor" type="BorderColorStyleWidthType"
            minOccurs="0" />
        <xsd:element name="BorderStyle" type="BorderColorStyleWidthType"
            minOccurs="0" />
        <xsd:element name="BorderWidth" type="BorderColorStyleWidthType"
            minOccurs="0" />
        <xsd:element name="BackgroundColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="BackgroundGradientType" type="xsd:string"
            minOccurs="0" />
        <xsd:element name="BackgroundGradientEndColor" type="xsd:string"
            minOccurs="0" />
        <xsd:element name="BackgroundImage" type="BackgroundImageType"
            minOccurs="0" />
        <xsd:element name="FontStyle" type="xsd:string" minOccurs="0" />
        <xsd:element name="FontFamily" type="xsd:string" minOccurs="0" />
        <xsd:element name="FontSize" type="xsd:string" minOccurs="0" />
        <xsd:element name="FontWeight" type="xsd:string" minOccurs="0" />
        <xsd:element name="Format" type="xsd:string" minOccurs="0" />
        <xsd:element name="TextDecoration" type="xsd:string" minOccurs="0" />
        <xsd:element name="TextAlign" type="xsd:string" minOccurs="0" />
        <xsd:element name="VerticalAlign" type="xsd:string" minOccurs="0" />
        <xsd:element name="Color" type="xsd:string" minOccurs="0" />
        <xsd:element name="PaddingLeft" type="xsd:string" minOccurs="0" />
        <xsd:element name="PaddingRight" type="xsd:string" minOccurs="0" />
        <xsd:element name="PaddingTop" type="xsd:string" minOccurs="0" />
        <xsd:element name="PaddingBottom" type="xsd:string" minOccurs="0" />
    </xsd:choice>

```

```

<xsd:element name="LineHeight" type="xsd:string" minOccurs="0" />
<xsd:element name="Direction" type="xsd:string" minOccurs="0" />
<xsd:element name="WritingMode" type="xsd:string" minOccurs="0" />
<xsd:element name="Language" type="xsd:string" minOccurs="0" />
<xsd:element name="UnicodeBiDi" type="xsd:string" minOccurs="0" />
<xsd:element name="Calendar" type="xsd:string" minOccurs="0" />
<xsd:element name="NumeralLanguage" type="xsd:string" minOccurs="0" />
<xsd:element name="NumeralVariant" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip"/>
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="BorderColorStyleWidthType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Default" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Right" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Bottom" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="BackgroundImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Source">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="External" />
          <xsd:enumeration value="Embedded" />
          <xsd:enumeration value="Database" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="MimeType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundRepeat" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="FiltersType">
  <xsd:sequence>
    <xsd:element name="Filter" type="FilterType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="FilterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="FilterExpression" type="xsd:string" />
    <xsd:element name="Operator">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Equal" />
          <xsd:enumeration value="Like" />
          <xsd:enumeration value="NotEqual" />
          <xsd:enumeration value="GreaterThan" />
          <xsd:enumeration value="GreaterThanOrEqual" />
          <xsd:enumeration value="LessThan" />
          <xsd:enumeration value="LessThanOrEqual" />
          <xsd:enumeration value="TopN" />
          <xsd:enumeration value="BottomN" />
          <xsd:enumeration value="TopPercent" />
          <xsd:enumeration value="BottomPercent" />
          <xsd:enumeration value="In" />
          <xsd:enumeration value="Between" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>

```

```

    <xsd:element name="FilterValues" type="FilterValuesType" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="FilterValuesType">
  <xsd:sequence>
    <xsd:element name="FilterValue" type="xsd:string" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="UserSortType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="SortExpression" type="xsd:string" />
    <xsd:element name="SortExpressionScope" type="xsd:string" minOccurs="0"/>
    <xsd:element name="SortTarget" type="xsd:string" minOccurs="0"/>
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:simpleType name="SizeType">
  <xsd:restriction base="xsd:normalizedString">
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="CustomPropertiesType">
  <xsd:sequence>
    <xsd:element name="CustomProperty" type="CustomPropertyType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CustomPropertyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" minOccurs="0"/>
    <xsd:element name="Value" type="xsd:string" />
    <xsd:any namespace="##other" processContents="skip"/>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
</xsd:schema>

```

### 5.3 RDL XML Schema for Version 2008/01

```

<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
  targetNamespace="http://schemas.microsoft.com/sqlserver/reporting/2008/01/reportdefinition"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.microsoft.com/sqlserver/reporting/2008/01/reportdefinition"
  elementFormDefault="qualified">
  <xsd:annotation>
    <xsd:documentation>

```

The following schema describes the structure of the Report Definition Language (RDL) for Microsoft SQL Server 2008.

THE SCHEMA IS PROVIDED TO YOU ON AN "AS IS" BASIS, AND MICROSOFT DISCLAIMS ALL WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT, AS TO THE SCHEMA OR ANY PRODUCT OR OTHER ITEM THAT MAY BE DEVELOPED USING THE SCHEMA.

Without limiting the generality of the foregoing, Microsoft makes no warranty that any product or other item that may be developed using the schema, or any portion of the schema, will not infringe any copyright, patent, trade secret or other intellectual property right of any individual or legal entity in any country. It is your responsibility to

obtain licenses to use any such intellectual property rights as appropriate.

MICROSOFT IS NOT LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING, WITHOUT LIMITATION, ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING LOST REVENUES OR LOST PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES.

(c) Microsoft Corporation. All rights reserved.

```
</xsd:documentation>
</xsd:annotation>
<xsd:element name="Report">
  <xsd:complexType>
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="Description" type="StringLocIDType" minOccurs="0" />
      <xsd:element name="Author" type="xsd:string" minOccurs="0" />
      <xsd:element name="AutoRefresh" type="xsd:unsignedInt" minOccurs="0" />
      <xsd:element name="DataSources" type="DataSourcesType" minOccurs="0" />
      <xsd:element name="DataSets" type="DataSetsType" minOccurs="0" />
      <xsd:element name="ReportParameters" type="ReportParametersType"
        minOccurs="0" />
      <xsd:element name="Code" type="xsd:string" minOccurs="0" />
      <xsd:element name="EmbeddedImages" type="EmbeddedImagesType"
        minOccurs="0" />
      <xsd:element name="Language" type="xsd:string" minOccurs="0" />
      <xsd:element name="CodeModules" type="CodeModulesType" minOccurs="0" />
      <xsd:element name="Classes" type="ClassesType" minOccurs="0" />
      <xsd:element name="CustomProperties" type="CustomPropertiesType"
        minOccurs="0" />
      <xsd:element name="Variables" type="VariablesType" minOccurs="0" />
      <xsd:element name="DeferVariableEvaluation" type="xsd:boolean"
        minOccurs="0" />
      <xsd:element name="ConsumeContainerWhitespace" type="xsd:boolean"
        minOccurs="0" />
      <xsd:element name="DataTransform" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataSchema" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataElementStyle" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Attribute" />
            <xsd:enumeration value="Element" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="Page" type="PageType" minOccurs="1" />
      <xsd:element name="Body" type="BodyType" minOccurs="1" />
      <xsd:element name="Width" type="SizeType" minOccurs="1" />
      <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>
</xsd:element>
<xsd:complexType name="ReportParametersType">
  <xsd:sequence>
    <xsd:element name="ReportParameter" type="ReportParameterType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ReportParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataType">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Boolean" />
          <xsd:enumeration value="DateTime" />
          <xsd:enumeration value="Integer" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
</xsd:complexType>
```

```

        <xsd:enumeration value="Float" />
        <xsd:enumeration value="String" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="Nullable" type="xsd:boolean" minOccurs="0" />
<xsd:element name="DefaultValue" type="DefaultValueType" minOccurs="0" />
<xsd:element name="AllowBlank" type="xsd:boolean" minOccurs="0" />
<xsd:element name="Prompt" type="StringLocIDType" minOccurs="0" />
<xsd:element name="ValidValues" type="ValidValuesType" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
<xsd:element name="MultiValue" type="xsd:boolean" minOccurs="0" />
<xsd:element name="UsedInQuery" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="False" />
            <xsd:enumeration value="True" />
            <xsd:enumeration value="Auto" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ValidValuesType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="DataSetReference" type="DataSetReferenceType"
            minOccurs="0" />
        <xsd:element name="ParameterValues" type="ParameterValuesType"
            minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataSetReferenceType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="DataSetName" type="xsd:string" />
        <xsd:element name="ValueField" type="xsd:string" />
        <xsd:element name="LabelField" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ParameterValuesType">
    <xsd:sequence>
        <xsd:element name="ParameterValue" type="ParameterValueType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ParameterValueType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Value" type="xsd:string" minOccurs="0" />
        <xsd:element name="Label" type="StringLocIDType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DefaultValueType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="DataSetReference" type="DataSetReferenceType"
            minOccurs="0" />
        <xsd:element name="Values" type="ValuesType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

<xsd:complexType name="ValueType">
  <xsd:sequence>
    <xsd:element name="Value" type="xsd:string" minOccurs="1"
      maxOccurs="unbounded" nillable="true" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataSetsType">
  <xsd:sequence>
    <xsd:element name="DataSet" type="DataSetType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataSetType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Fields" type="FieldsType" minOccurs="0" />
    <xsd:element name="Query" type="QueryType" />
    <xsd:element name="CaseSensitivity" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="True" />
          <xsd:enumeration value="False" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Collation" type="xsd:string" minOccurs="0" />
    <xsd:element name="AccentSensitivity" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="True" />
          <xsd:enumeration value="False" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="KanatypeSensitivity" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="True" />
          <xsd:enumeration value="False" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="WidthSensitivity" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="True" />
          <xsd:enumeration value="False" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="InterpretSubtotalsAsDetails" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="True" />
          <xsd:enumeration value="False" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```



```

<xsd:complexType name="FieldsType">
  <xsd:sequence>
    <xsd:element name="Field" type="FieldType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="StringWithDataModelAttribute">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="DataType" use="optional">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Boolean" />
            <xsd:enumeration value="DateTime" />
            <xsd:enumeration value="Integer" />
            <xsd:enumeration value="Float" />
            <xsd:enumeration value="String" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:anyAttribute namespace="##other" processContents="skip" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="FieldType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataField" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="StringWithDataModelAttribute"
      minOccurs="0" />
  <xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="QueryType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataSourceName" type="xsd:string" />
    <xsd:element name="CommandType" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Text" />
          <xsd:enumeration value="StoredProcedure" />
          <xsd:enumeration value="TableDirect" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="CommandText" type="xsd:string" />
    <xsd:element name="QueryParameters" type="QueryParametersType"
      minOccurs="0" />
    <xsd:element name="Timeout" type="xsd:unsignedInt" minOccurs="0" />
  <xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataSourcesType">
  <xsd:sequence>
    <xsd:element name="DataSource" type="DataSourceType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataSourceType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Transaction" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ConnectionProperties" type="ConnectionPropertiesType"
      minOccurs="0" />
    <xsd:element name="DataSourceReference" type="xsd:string" minOccurs="0" />
  <xsd:any namespace="##other" processContents="skip" />
</xsd:choice>

```

```

    <xsd:attribute name="Name" type="xsd:string" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ConnectionPropertiesType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataProvider" type="xsd:string" />
    <xsd:element name="ConnectString" type="xsd:string" />
    <xsd:element name="IntegratedSecurity" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Prompt" type="StringLocIDType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="QueryParametersType">
  <xsd:sequence>
    <xsd:element name="QueryParameter" type="QueryParameterType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="QueryParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="StringWithDataAttribute" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CodeModulesType">
  <xsd:sequence>
    <xsd:element name="CodeModule" type="xsd:string" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ClassesType">
  <xsd:sequence>
    <xsd:element name="Class" type="ClassType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ClassType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ClassName" type="xsd:string" />
    <xsd:element name="InstanceName" type="xsd:normalizedString" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="BodyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="1" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="PageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="PageHeader" type="PageSectionType" minOccurs="0" />
    <xsd:element name="PageFooter" type="PageSectionType" minOccurs="0" />
    <xsd:element name="PageHeight" type="SizeType" minOccurs="0" />
    <xsd:element name="PageWidth" type="SizeType" minOccurs="0" />
    <xsd:element name="InteractiveHeight" type="SizeType" minOccurs="0" />
    <xsd:element name="InteractiveWidth" type="SizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="RightMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="TopMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="BottomMargin" type="SizeType" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:element name="Columns" type="xsd:int" minOccurs="0" />
    <xsd:element name="ColumnSpacing" type="SizeType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="PageSectionType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="PrintOnFirstPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PrintOnLastPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PrintBetweenSections" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="EmbeddedImagesType">
  <xsd:sequence>
    <xsd:element name="EmbeddedImage" type="EmbeddedImageType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="EmbeddedImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="MIMEType" type="xsd:string" />
    <xsd:element name="ImageData" type="xsd:string" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ReportItemsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Line" type="LineType" />
    <xsd:element name="Rectangle" type="RectangleType" />
    <xsd:element name="Textbox" type="TextboxType" />
    <xsd:element name="Image" type="ImageType" />
    <xsd:element name="Subreport" type="SubreportType" />
    <xsd:element name="Chart" type="ChartType" />
    <xsd:element name="GaugePanel" type="GaugePanelType" />
    <xsd:element name="Tablix" type="TablixType" />
    <xsd:element name="CustomReportItem" type="CustomReportItemType" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ActionInfoType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Actions" type="ActionsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ActionsType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Action" type="ActionType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ActionType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hyperlink" type="xsd:string" minOccurs="0" />
    <xsd:element name="Drillthrough" type="DrillthroughType" minOccurs="0" />

```

```

    <xsd:element name="BookmarkLink" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DrillthroughType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportName" type="xsd:string" />
    <xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="VisibilityType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToggleItem" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="LineType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="RectangleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"

```

```

        minOccurs="0" />
<xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
<xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean"
    minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Output" />
            <xsd:enumeration value="NoOutput" />
            <xsd:enumeration value="ContentsOnly" />
            <xsd:enumeration value="Auto" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TextboxType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:element name="Top" type="SizeType" minOccurs="0" />
        <xsd:element name="Left" type="SizeType" minOccurs="0" />
        <xsd:element name="Height" type="SizeType" minOccurs="0" />
        <xsd:element name="Width" type="SizeType" minOccurs="0" />
        <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
        <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
        <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
        <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType"
            minOccurs="0" />
        <xsd:element name="Paragraphs" type="ParagraphsType" minOccurs="1" />
        <xsd:element name="CanGrow" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="CanShrink" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="HideDuplicates" type="xsd:string" minOccurs="0" />
        <xsd:element name="ToggleImage" type="ToggleImageType" minOccurs="0" />
        <xsd:element name="UserSort" type="UserSortType" minOccurs="0" />
        <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
        <xsd:element name="DataElementOutput" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Output" />
                    <xsd:enumeration value="NoOutput" />
                    <xsd:enumeration value="ContentsOnly" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="DataElementStyle" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Auto" />
                    <xsd:enumeration value="Attribute" />
                    <xsd:enumeration value="Element" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />

```

```

</xsd:complexType>
<xsd:complexType name="ParagraphsType">
  <xsd:sequence>
    <xsd:element name="Paragraph" type="ParagraphType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ParagraphType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TextRuns" type="TextRunsType" minOccurs="1" />
    <xsd:element name="LeftIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="HangingIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="ListStyle" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Bulleted" />
          <xsd:enumeration value="Numbered" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="ListLevel" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="SpaceBefore" type="xsd:string" minOccurs="0" />
    <xsd:element name="SpaceAfter" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TextRunsType">
  <xsd:sequence>
    <xsd:element name="TextRun" type="TextRunType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TextRunType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Value" type="LocIDStringWithDataAttribute"
      minOccurs="1" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkupType" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ToggleImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="InitialState" type="xsd:string" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType"
  minOccurs="0" />
<xsd:element name="Source">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="External" />
      <xsd:enumeration value="Embedded" />
      <xsd:enumeration value="Database" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="Value" type="xsd:string" />
<xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
<xsd:element name="Sizing" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="AutoSize" />
      <xsd:enumeration value="Fit" />
      <xsd:enumeration value="FitProportional" />
      <xsd:enumeration value="Clip" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="SubreportType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="ReportName" type="xsd:string" />
    <xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
    <xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
    <xsd:element name="MergeTransactions" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean"
      minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">

```

```

        <xsd:enumeration value="Output" />
        <xsd:enumeration value="NoOutput" />
        <xsd:enumeration value="ContentsOnly" />
        <xsd:enumeration value="Auto" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CustomReportItemType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Type" type="xsd:string" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:element name="Top" type="SizeType" minOccurs="0" />
        <xsd:element name="Left" type="SizeType" minOccurs="0" />
        <xsd:element name="Height" type="SizeType" minOccurs="0" />
        <xsd:element name="Width" type="SizeType" minOccurs="0" />
        <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
        <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
        <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
        <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType"
            minOccurs="0" />
        <xsd:element name="AltReportItem" type="ReportItemsType" minOccurs="0" />
        <xsd:element name="CustomData" type="CustomDataType" minOccurs="0" />
        <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
        <xsd:element name="DataElementOutput" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Output" />
                    <xsd:enumeration value="NoOutput" />
                    <xsd:enumeration value="ContentsOnly" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CustomDataType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="DataSetName" type="xsd:string" />
        <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
        <xsd:element name="SortExpressions" type="SortExpressionsType"
            minOccurs="0" />
        <xsd:element name="DataColumnHierarchy" type="DataColumnHierarchyType"
            minOccurs="0" />
        <xsd:element name="DataRowHierarchy" type="DataRowHierarchyType"
            minOccurs="0" />
        <xsd:element name="DataRows" type="DataRowsType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataColumnHierarchyType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="DataMembers" type="DataMembersType" />
        <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```



```

<xsd:complexType name="DataRowHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataMembers" type="DataMembersType" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataMembersType">
  <xsd:sequence>
    <xsd:element name="DataMember" type="DataMemberType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataMemberType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType"
      minOccurs="0" />
    <xsd:element name="Subtotal" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="DataMembers" type="DataMembersType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataRowsType">
  <xsd:sequence>
    <xsd:element name="DataRow" type="DataRowType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataRowType">
  <xsd:sequence>
    <xsd:element name="DataCell" type="DataCellType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataCellType">
  <xsd:sequence>
    <xsd:element name="DataValue" type="DataValueType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ParametersType">
  <xsd:sequence>
    <xsd:element name="Parameter" type="ParameterType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="Omit" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="GroupType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="GroupExpressions" type="GroupExpressionsType"
      minOccurs="0" />
    <xsd:element name="ReGroupExpressions" type="GroupExpressionsType"
      minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="Parent" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="Variables" type="VariablesType" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="VariablesType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Variable" type="VariableType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="VariableType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="StringWithDataModelAttribute" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="GroupExpressionsType">
  <xsd:sequence>
    <xsd:element name="GroupExpression" type="xsd:string"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="SortExpressionsType">
  <xsd:sequence>
    <xsd:element name="SortExpression" type="SortExpressionType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="SortExpressionType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="Direction" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Ascending" />
          <xsd:enumeration value="Descending" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType"
      minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="Top" type="SizeType" minOccurs="0" />
<xsd:element name="Left" type="SizeType" minOccurs="0" />
<xsd:element name="Height" type="SizeType" minOccurs="0" />
<xsd:element name="Width" type="SizeType" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType"
minOccurs="0" />
<xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:element name="ChartSeriesHierarchy" type="ChartHierarchyType" />
<xsd:element name="ChartCategoryHierarchy" type="ChartHierarchyType" />
<xsd:element name="ChartData" type="ChartDataTypes" minOccurs="0" />
<xsd:element name="ChartAreas" type="ChartAreasType" minOccurs="0" />
<xsd:element name="ChartLegends" type="ChartLegendsType" minOccurs="0" />
<xsd:element name="ChartTitles" type="ChartTitlesType" minOccurs="0" />
<xsd:element name="DynamicHeight" type="xsd:string" minOccurs="0" />
<xsd:element name="DynamicWidth" type="xsd:string" minOccurs="0" />
<xsd:element name="Palette" type="xsd:string" minOccurs="0" />
<xsd:element name="ChartCustomPaletteColors"
type="ChartCustomPaletteColorsType" minOccurs="0" />
<xsd:element name="PaletteHatchBehavior" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="ChartBorderSkin" type="ChartBorderSkinType"
minOccurs="0" />
<xsd:element name="Code" type="xsd:string" minOccurs="0" />
<xsd:element name="CodeLanguage" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="CSharp" />
      <xsd:enumeration value="VB" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="ChartCodeParameters" type="ChartCodeParametersType"
minOccurs="0" />
<xsd:element name="ChartAnnotations" type="ChartAnnotationsType"
minOccurs="0" />
<xsd:element name="ChartNoDataMessage" type="ChartTitleType"
minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ChartMembers" type="ChartMembersType" minOccurs="1"
maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

<xsd:complexType name="ChartMembersType">
  <xsd:sequence minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ChartMember" type="ChartMemberType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartMemberType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="ChartMembers" type="ChartMembersType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="Label" type="StringLocIDType" minOccurs="1"
      maxOccurs="1" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" maxOccurs="1" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0" maxOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartAreasType">
  <xsd:sequence>
    <xsd:element name="ChartArea" type="ChartAreaType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartAreaType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartCategoryAxes" type="ChartCategoryAxesType"
      minOccurs="0" />
    <xsd:element name="ChartValueAxes" type="ChartValueAxesType"
      minOccurs="0" />
    <xsd:element name="ChartThreeDProperties" type="ChartThreeDPropertiesType"
      minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="AlignOrientation" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartAlignType" type="ChartAlignTypeType"
      minOccurs="0" />
    <xsd:element name="ChartElementPosition" type="ChartElementPositionType"
      minOccurs="0" />
    <xsd:element name="ChartInnerPlotPosition" type="ChartElementPositionType"
      minOccurs="0" />
    <xsd:element name="AlignWithChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="EquallySizedAxesFont" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartAlignTypeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="AxesView" type="xsd:string" minOccurs="0" />
    <xsd:element name="Cursor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="InnerPlotPosition" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartElementPositionType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartTitlesType">
  <xsd:sequence>
    <xsd:element name="ChartTitle" type="ChartTitleType" minOccurs="0"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartTitleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Caption" type="StringLocIDType" minOccurs="1" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockToChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartElementPosition" type="ChartElementPositionType"
      minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartLegendsType">
  <xsd:sequence>
    <xsd:element name="ChartLegend" type="ChartLegendType" minOccurs="0"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartLegendType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="Layout" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockToChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartElementPosition" type="ChartElementPositionType"
      minOccurs="0" />
    <xsd:element name="ChartLegendTitle" type="ChartLegendTitleType"
      minOccurs="0" />
    <xsd:element name="AutoFitTextDisabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartLegendColumns" type="ChartLegendColumnsType"
      minOccurs="0" />
    <xsd:element name="HeaderSeparator" type="xsd:string" minOccurs="0" />
    <xsd:element name="HeaderSeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ColumnSeparator" type="xsd:string" minOccurs="0" />
    <xsd:element name="ColumnSeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ColumnSpacing" type="xsd:string" minOccurs="0" />
    <xsd:element name="InterlacedRows" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:element name="InterlacedRowsColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="EquallySpacedItems" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaxAutoSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextWrapThreshold" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartLegendTitleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Caption" type="StringLocIDType" />
    <xsd:element name="TitleSeparator" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartCustomPaletteColorsType">
  <xsd:sequence>
    <xsd:element name="ChartCustomPaletteColor" type="xsd:string"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartBorderSkinType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartBorderSkinType" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="ChartCodeParametersType">
  <xsd:sequence>
    <xsd:element name="ChartCodeParameter" type="ChartCodeParameterType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartCodeParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartLegendColumnsType">
  <xsd:sequence>
    <xsd:element name="ChartLegendColumn" type="ChartLegendColumnType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartLegendColumnType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ColumnType">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Text" />
          <xsd:enumeration value="SeriesSymbol" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:element name="MinimumWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeriesSymbolWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeriesSymbolHeight" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartLegendColumnHeaderType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartLegendCustomItemsType">
  <xsd:sequence>
    <xsd:element name="ChartLegendCustomItem" type="ChartLegendCustomItemType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartLegendCustomItemType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartLegendCustomItemCells"
      type="ChartLegendCustomItemCellsType" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
    <xsd:element name="Separator" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartLegendCustomItemCellsType">
  <xsd:sequence>
    <xsd:element name="ChartLegendCustomItemCell"
      type="ChartLegendCustomItemCellType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartLegendCustomItemCellType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="CellType" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Text" />
          <xsd:enumeration value="SeriesSymbol" />
          <xsd:enumeration value="Image" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="CellSpan" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ImageHeight" type="xsd:string" minOccurs="0" />
    <xsd:element name="ImageWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="SymbolHeight" type="xsd:string" minOccurs="0" />
    <xsd:element name="SymbolWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="Alignment" type="xsd:string" minOccurs="0" />
    <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartMarkerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <xsd:element name="Size" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartCategoryAxesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartAxis" type="ChartAxisType" maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartValueAxesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartAxis" type="ChartAxisType" minOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartAxisType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Visible" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartAxisTitle" type="ChartAxisTitleType"
      minOccurs="0" />
    <xsd:element name="Margin" type="xsd:string" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:element name="VariableAutoInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelIntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelIntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelIntervalOffsetType" type="xsd:string"
      minOccurs="0" />
    <xsd:element name="ChartMajorGridLines" type="ChartGridLinesType"
      minOccurs="0" />
    <xsd:element name="ChartMinorGridLines" type="ChartGridLinesType"
      minOccurs="0" />
    <xsd:element name="ChartMajorTickMarks" type="ChartTickMarksType"
      minOccurs="0" />
    <xsd:element name="ChartMinorTickMarks" type="ChartTickMarksType"
      minOccurs="0" />
    <xsd:element name="MarksAlwaysAtPlotEdge" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reverse" type="xsd:string" minOccurs="0" />
    <xsd:element name="CrossAt" type="xsd:string" minOccurs="0" />
    <xsd:element name="Location" type="xsd:string" minOccurs="0" />
    <xsd:element name="Interlaced" type="xsd:string" minOccurs="0" />
    <xsd:element name="InterlacedColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartStripLines" type="ChartStripLinesType"
      minOccurs="0" />
    <xsd:element name="Arrows" type="xsd:string" minOccurs="0" />
    <xsd:element name="Scalar" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Minimum" type="xsd:string" minOccurs="0" />
    <xsd:element name="Maximum" type="xsd:string" minOccurs="0" />
    <xsd:element name="LogScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="LogBase" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```



```

<xsd:element name="HideLabels" type="xsd:string" minOccurs="0" />
<xsd:element name="Angle" type="xsd:string" minOccurs="0" />
<xsd:element name="PreventFontShrink" type="xsd:string" minOccurs="0" />
<xsd:element name="PreventFontGrow" type="xsd:string" minOccurs="0" />
<xsd:element name="PreventLabelOffset" type="xsd:string" minOccurs="0" />
<xsd:element name="PreventWordWrap" type="xsd:string" minOccurs="0" />
<xsd:element name="AllowLabelRotation" type="xsd:string" minOccurs="0" />
<xsd:element name="IncludeZero" type="xsd:string" minOccurs="0" />
<xsd:element name="LabelsAutoFitDisabled" type="xsd:string" minOccurs="0" />
<xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
<xsd:element name="MaxFontSize" type="xsd:string" minOccurs="0" />
<xsd:element name="OffsetLabels" type="xsd:string" minOccurs="0" />
<xsd:element name="HideEndLabels" type="xsd:string" minOccurs="0" />
<xsd:element name="ChartAxisScaleBreak" type="ChartAxisScaleBreakType"
  minOccurs="0" />
  <xsd:element name="CustomProperties" type="CustomPropertiesType"
    minOccurs="0" />
  <xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartAxisTitleType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Caption" type="StringLocIDType" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartStripLinesType">
  <xsd:sequence>
    <xsd:element name="ChartStripLine" type="ChartStripLineType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartStripLineType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Title" type="xsd:string" minOccurs="0" />
    <xsd:element name="TitleAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:element name="StripWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="StripWidthType" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartAxisScaleBreakType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="BreakLineType" type="xsd:string" minOccurs="0" />
    <xsd:element name="CollapsibleSpaceThreshold" type="xsd:string"
      minOccurs="0" />
    <xsd:element name="MaxNumberOfBreaks" type="xsd:string" minOccurs="0" />
    <xsd:element name="Spacing" type="xsd:string" minOccurs="0" />
    <xsd:element name="IncludeZero" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>

```

```

    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartDataTypes">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="ChartSeriesCollection" type="ChartSeriesCollectionType"
      minOccurs="1" maxOccurs="1" />
    <xsd:element name="ChartDerivedSeriesCollection"
      type="ChartDerivedSeriesCollectionType" minOccurs="0" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartSeriesCollectionType">
  <xsd:sequence maxOccurs="unbounded" minOccurs="1">
    <xsd:element name="ChartSeries" type="ChartSeriesType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartDerivedSeriesCollectionType">
  <xsd:sequence maxOccurs="unbounded" minOccurs="1">
    <xsd:element name="ChartDerivedSeries" type="ChartDerivedSeriesType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartSeriesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartDataPoints" type="ChartDataPointsType"
      minOccurs="0" />
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <xsd:element name="Subtype" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartEmptyPoints" type="ChartEmptyPointsType"
      minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartItemInLegend" type="ChartItemInLegendType"
      minOccurs="0" />
    <xsd:element name="ChartAreaName" type="xsd:string" minOccurs="0" />
    <xsd:element name="ValueAxisName" type="xsd:string" minOccurs="0" />
    <xsd:element name="CategoryAxisName" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartSmartLabel" type="ChartSmartLabelType"
      minOccurs="0" />
    <xsd:element name="ChartDataLabel" type="ChartDataLabelType"
      minOccurs="0" />
    <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
  <xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartDerivedSeriesType">
  <xsd:choice minOccurs="3" maxOccurs="unbounded">
    <xsd:element name="ChartSeries" type="ChartSeriesType" minOccurs="1" />
    <xsd:element name="SourceChartSeriesName" type="xsd:string" minOccurs="1" />
    <xsd:element name="DerivedSeriesFormula" minOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="RuningTotal" />
          <xsd:enumeration value="RuningAverage" />
          <xsd:enumeration value="MoneyFlow" />
          <xsd:enumeration value="OnBalanceVolume" />
          <xsd:enumeration value="NegativeVolumeIndex" />
          <xsd:enumeration value="PositiveVolumeIndex" />
          <xsd:enumeration value="PriceVolumeTrend" />
          <xsd:enumeration value="AccumulationDistribution" />
          <xsd:enumeration value="Forecasting" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

<xsd:enumeration value="StandardDeviation" />
<xsd:enumeration value="AverageTruerance" />
<xsd:enumeration value="EaseOfMovement" />
<xsd:enumeration value="MassIndex" />
<xsd:enumeration value="Performance" />
<xsd:enumeration value="RateOfChange" />
<xsd:enumeration value="RelativeStrengthIndex" />
<xsd:enumeration value="TRIX" />
<xsd:enumeration value="MACD" />
<xsd:enumeration value="CommodityChannelIndex" />
<xsd:enumeration value="TTestEqualVariances" />
<xsd:enumeration value="TTestUnequalVariances" />
<xsd:enumeration value="TTestPaired" />
<xsd:enumeration value="ZTest" />
<xsd:enumeration value="FTest" />
<xsd:enumeration value="Covariance" />
<xsd:enumeration value="Correlation" />
<xsd:enumeration value="Anova" />
<xsd:enumeration value="TDistribution" />
<xsd:enumeration value="FDistribution" />
<xsd:enumeration value="NormalDistribution" />
<xsd:enumeration value="InverseTDistribution" />
<xsd:enumeration value="InverseFDistribution" />
<xsd:enumeration value="InverseNormalDistribution" />
<xsd:enumeration value="Mean" />
<xsd:enumeration value="Variance" />
<xsd:enumeration value="Median" />
<xsd:enumeration value="BetaFunction" />
<xsd:enumeration value="GammaFunction" />
<xsd:enumeration value="MovingAverage" />
<xsd:enumeration value="ExponentialMovingAverage" />
<xsd:enumeration value="TriangularMovingAverage" />
<xsd:enumeration value="WeightedMovingAverage" />
<xsd:enumeration value="BollingerBands" />
<xsd:enumeration value="MedianPrice" />
<xsd:enumeration value="TypicalPrice" />
<xsd:enumeration value="WeightedClose" />
<xsd:enumeration value="Envelopes" />
<xsd:enumeration value="StochasticIndicator" />
<xsd:enumeration value="ChaikinOscillator" />
<xsd:enumeration value="DetrendedPriceOscillator" />
<xsd:enumeration value="VolatilityChankins" />
<xsd:enumeration value="VolumeOscillator" />
<xsd:enumeration value="WilliansR" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="ChartFormulaParameters" type="ChartFormulaParametersType"
  minOccurs="0" />
  <xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartFormulaParametersType">
  <xsd:sequence>
    <xsd:element name="ChartFormulaParameter" type="ChartFormulaParameterType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartFormulaParameterType">
  <xsd:choice minOccurs="1">
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Source" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

<xsd:complexType name="ChartEmptyPointsType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
    <xsd:element name="ChartDataLabel" type="ChartDataLabelType"
      minOccurs="0" />
    <xsd:element name="AxisLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartItemInLegendType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartDataPointsType">
  <xsd:sequence>
    <xsd:element name="ChartDataPoint" type="ChartDataPointType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartDataPointType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartDataPointValues" type="ChartDataPointValuesType"
      minOccurs="0" />
    <xsd:element name="ChartDataLabel" type="ChartDataLabelType"
      minOccurs="0" />
    <xsd:element name="AxisLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
    <xsd:element name="ChartItemInLegend" type="ChartItemInLegendType"
      minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartDataPointValuesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="X" type="xsd:string" minOccurs="0" />
    <xsd:element name="Y" type="xsd:string" minOccurs="0" />
    <xsd:element name="Size" type="xsd:string" minOccurs="0" />
    <xsd:element name="High" type="xsd:string" minOccurs="0" />
    <xsd:element name="Low" type="xsd:string" minOccurs="0" />
    <xsd:element name="Start" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:element name="End" type="xsd:string" minOccurs="0" />
    <xsd:element name="Mean" type="xsd:string" minOccurs="0" />
    <xsd:element name="Median" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="DataValueType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartDataLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Visible" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Label" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="UseValueAsLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="Rotation" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartSmartLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Disabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowOutsidePlotArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutBackColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineAnchor" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowOverlapped" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerOverlapping" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaxMovingDistance" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinMovingDistance" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartNoMoveDirections" type="ChartNoMoveDirectionsType"
      minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartNoMoveDirectionsType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Up" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Right" type="xsd:string" minOccurs="0" />
    <xsd:element name="Down" type="xsd:string" minOccurs="0" />
    <xsd:element name="UpLeft" type="xsd:string" minOccurs="0" />
    <xsd:element name="UpRight" type="xsd:string" minOccurs="0" />
    <xsd:element name="DownLeft" type="xsd:string" minOccurs="0" />
    <xsd:element name="DownRight" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartThreeDPropertiesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="ProjectionMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="Rotation" type="xsd:string" minOccurs="0" />
    <xsd:element name="Inclination" type="xsd:string" minOccurs="0" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

    <xsd:element name="Perspective" type="xsd:string" minOccurs="0" />
    <xsd:element name="DepthRatio" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shading" type="xsd:string" minOccurs="0" />
    <xsd:element name="GapDepth" type="xsd:string" minOccurs="0" />
    <xsd:element name="WallThickness" type="xsd:string" minOccurs="0" />
    <xsd:element name="Clustered" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartGridLinesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartTickMarksType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartAnnotationsType">
  <xsd:sequence>
    <xsd:element name="ChartAnnotation" type="ChartAnnotationType"
      maxOccurs="unbounded" />
    <!--TODO-->
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ChartAnnotationType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--TODO-->
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="StyleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Border" type="BorderType" minOccurs="0" />
    <xsd:element name="TopBorder" type="BorderType" minOccurs="0" />
    <xsd:element name="BottomBorder" type="BorderType" minOccurs="0" />
    <xsd:element name="LeftBorder" type="BorderType" minOccurs="0" />
    <xsd:element name="RightBorder" type="BorderType" minOccurs="0" />
    <xsd:element name="BackgroundColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundGradientType" type="xsd:string"
      minOccurs="0" />
    <xsd:element name="BackgroundGradientEndColor" type="xsd:string"
      minOccurs="0" />
    <xsd:element name="BackgroundHatchType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundImage" type="BackgroundImageType"
      minOccurs="0" />
    <xsd:element name="FontStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontFamily" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontWeight" type="xsd:string" minOccurs="0" />
    <xsd:element name="Format" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="TextDecoration" type="xsd:string" minOccurs="0" />
<xsd:element name="TextAlign" type="xsd:string" minOccurs="0" />
<xsd:element name="TextEffect" type="xsd:string" minOccurs="0" />
<xsd:element name="VerticalAlign" type="xsd:string" minOccurs="0" />
<xsd:element name="Color" type="xsd:string" minOccurs="0" />
<xsd:element name="ShadowColor" type="xsd:string" minOccurs="0" />
<xsd:element name="ShadowOffset" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingLeft" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingRight" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingTop" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingBottom" type="xsd:string" minOccurs="0" />
<xsd:element name="LineHeight" type="xsd:string" minOccurs="0" />
<xsd:element name="Direction" type="xsd:string" minOccurs="0" />
<xsd:element name="WritingMode" type="xsd:string" minOccurs="0" />
<xsd:element name="Language" type="xsd:string" minOccurs="0" />
<xsd:element name="UnicodeBiDi" type="xsd:string" minOccurs="0" />
<xsd:element name="Calendar" type="xsd:string" minOccurs="0" />
<xsd:element name="NumeralLanguage" type="xsd:string" minOccurs="0" />
<xsd:element name="NumeralVariant" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="BorderStyle">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Color" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="BackgroundImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Source">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="External" />
          <xsd:enumeration value="Embedded" />
          <xsd:enumeration value="Database" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundRepeat" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="FiltersType">
  <xsd:sequence>
    <xsd:element name="Filter" type="FilterType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="FilterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="FilterExpression" type="xsd:string" />
    <xsd:element name="Operator">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Equal" />
          <xsd:enumeration value="Like" />
          <xsd:enumeration value="NotEqual" />
          <xsd:enumeration value="GreaterThan" />
          <xsd:enumeration value="GreaterThanOrEqual" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>

```

```

        <xsd:enumeration value="LessThan" />
        <xsd:enumeration value="LessThanOrEqual" />
        <xsd:enumeration value="TopN" />
        <xsd:enumeration value="BottomN" />
        <xsd:enumeration value="TopPercent" />
        <xsd:enumeration value="BottomPercent" />
        <xsd:enumeration value="In" />
        <xsd:enumeration value="Between" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="FilterValues" type="FilterValuesType" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="FilterValuesType">
    <xsd:sequence>
        <xsd:element name="FilterValue" type="StringWithDataModelAttribute"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="UserSortType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="SortExpression" type="xsd:string" />
        <xsd:element name="SortExpressionScope" type="xsd:string" minOccurs="0" />
        <xsd:element name="SortTarget" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:simpleType name="SizeType">
    <xsd:restriction base="xsd:normalizedString">
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="StringLocIDType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:anyAttribute namespace="##other" processContents="skip" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="LocIDStringWithDataModelAttribute">
    <xsd:simpleContent>
        <xsd:extension base="StringWithDataModelAttribute">
            <xsd:attribute name="EvaluationMode" type="EvaluationModeType"
                default="Auto" />
            <xsd:anyAttribute namespace="##other" processContents="skip" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="EvaluationModeType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Auto" />
        <xsd:enumeration value="Constant" />
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="CustomPropertiesType">
    <xsd:sequence>
        <xsd:element name="CustomProperty" type="CustomPropertyType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CustomPropertyType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Name" type="xsd:string" />
        <xsd:element name="Value" type="xsd:string" />
    </xsd:choice>
</xsd:complexType>

```



```

    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TablixType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixCorner" type="TablixCornerType" minOccurs="0" />
    <xsd:element name="TablixBody" type="TablixBodyType" minOccurs="0" />
    <xsd:element name="TablixColumnHierarchy" type="TablixHierarchyType"
      minOccurs="1" />
    <xsd:element name="TablixRowHierarchy" type="TablixHierarchyType"
      minOccurs="1" />
    <xsd:element name="LayoutDirection" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="LTR" />
          <xsd:enumeration value="RTL" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="GroupsBeforeRowHeaders" type="xsd:unsignedInt"
      minOccurs="0" />
    <xsd:element name="RepeatColumnHeaders" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="RepeatRowHeaders" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="FixedColumnHeaders" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="FixedRowHeaders" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType"
      minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean"
      minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TablixBodyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixColumns" type="TablixColumnsType" minOccurs="1"
      maxOccurs="1" />

```

```

    <xsd:element name="TablixRows" type="TablixRowsType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="TablixCornerType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixCornerRows" type="TablixCornerRowsType"
      minOccurs="1" maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TablixCornerRowsType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixCornerRow" type="TablixCornerRowType" minOccurs="1"
      maxOccurs="unbounded">
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TablixCornerRowType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="TablixCornerCell" type="TablixCornerCellType"
      minOccurs="0" maxOccurs="unbounded">
    </xsd:element>
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TablixCornerCellType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="CellContents" type="CellContentsType" minOccurs="0"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TablixHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TablixMembersType">
  <xsd:sequence minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixMember" type="TablixMemberType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TablixMemberType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="TablixHeader" type="TablixHeaderType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType"
      minOccurs="0" maxOccurs="1" />
    <xsd:element name="FixedData" type="xsd:boolean" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0"

```

```

        maxOccurs="1" />
<xsd:element name="HideIfNoRows" type="xsd:boolean" minOccurs="0" />
<xsd:element name="RepeatOnNewPage" type="xsd:boolean" minOccurs="0" />
<xsd:element name="KeepWithGroup" minOccurs="0" maxOccurs="1">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="None" />
      <xsd:enumeration value="Before" />
      <xsd:enumeration value="After" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TablixHeaderType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Size" type="SizeType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="CellContents" type="CellContentsType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CellContentsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="ColSpan" type="xsd:unsignedInt" minOccurs="0"
      maxOccurs="1">
    </xsd:element>
    <xsd:element name="RowSpan" type="xsd:unsignedInt" minOccurs="0"
      maxOccurs="1">
    </xsd:element>
    <xsd:element name="Line" type="LineType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="Rectangle" type="RectangleType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="Textbox" type="TextboxType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="Image" type="ImageType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="Subreport" type="SubreportType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="Chart" type="ChartType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="GaugePanel" type="GaugePanelType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="CustomReportItem" type="CustomReportItemType"
      minOccurs="0" maxOccurs="1" />
    <xsd:element name="Tablix" type="TablixType" minOccurs="0" maxOccurs="1" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TablixColumnsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="TablixColumn" type="TablixColumnType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>

```

```

    <xsd:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>
  <xsd:complexType name="TablixColumnType">
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="Width" type="SizeType" minOccurs="1" maxOccurs="1" />
      <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>
  <xsd:complexType name="TablixRowsType">
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="TablixRow" type="TablixRowType" minOccurs="1"
        maxOccurs="unbounded" />
      <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>
  <xsd:complexType name="TablixRowType">
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="Height" type="SizeType" minOccurs="1" maxOccurs="1" />
      <xsd:element name="TablixCells" type="TablixCellsType" minOccurs="1"
        maxOccurs="1" />
      <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>
  <xsd:complexType name="TablixCellsType">
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="TablixCell" type="TablixCellType" minOccurs="1"
        maxOccurs="unbounded" />
      <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>
  <xsd:complexType name="TablixCellType">
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="CellContents" type="CellContentsType" minOccurs="0"
        maxOccurs="1" />
      <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"
        maxOccurs="1" />
      <xsd:element name="DataElementOutput" minOccurs="0" maxOccurs="1">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Output" />
            <xsd:enumeration value="NoOutput" />
            <xsd:enumeration value="ContentsOnly" />
            <xsd:enumeration value="Auto" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>
  <xsd:complexType name="PageBreakType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="BreakLocation" minOccurs="1">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="None" />
            <xsd:enumeration value="Start" />
            <xsd:enumeration value="End" />
            <xsd:enumeration value="StartAndEnd" />
            <xsd:enumeration value="Between" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
  </xsd:complexType>

```

```

    <xsd:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>
  <xsd:complexType name="GaugePanelType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <!--DataRegionTypeStart-->
      <xsd:element name="Style" type="StyleType" minOccurs="0" />
      <xsd:element name="SortExpressions" type="SortExpressionsType"
        minOccurs="0" />
      <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
      <xsd:element name="Top" type="SizeType" minOccurs="0" />
      <xsd:element name="Left" type="SizeType" minOccurs="0" />
      <xsd:element name="Height" type="SizeType" minOccurs="0" />
      <xsd:element name="Width" type="SizeType" minOccurs="0" />
      <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
      <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
      <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
      <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
      <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
      <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
      <xsd:element name="CustomProperties" type="CustomPropertiesType"
        minOccurs="0" />
      <xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
      <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
      <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
      <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataElementOutput" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Output" />
            <xsd:enumeration value="NoOutput" />
            <xsd:enumeration value="ContentsOnly" />
            <xsd:enumeration value="Auto" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <!--DataRegionTypeEnd-->
      <xsd:element name="AntiAliasing" type="xsd:string" minOccurs="0" />
      <xsd:element name="TextAntiAliasingQuality" type="xsd:string"
        minOccurs="0" />
      <xsd:element name="AutoLayout" type="xsd:string" minOccurs="0" />
      <xsd:element name="ShadowIntensity" type="xsd:string" minOccurs="0" />
      <xsd:element name="RadialGauges" type="RadialGaugesType" minOccurs="0" />
      <xsd:element name="LinearGauges" type="LinearGaugesType" minOccurs="0" />
      <xsd:element name="NumericIndicators" type="NumericIndicatorsType"
        minOccurs="0" />
      <xsd:element name="StateIndicators" type="StateIndicatorsType"
        minOccurs="0" />
      <xsd:element name="GaugeImages" type="GaugeImagesType" minOccurs="0" />
      <xsd:element name="GaugeLabels" type="GaugeLabelsType" minOccurs="0" />
      <xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
      <xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
      <xsd:element name="GaugeMember" type="GaugeMemberType" minOccurs="0" />
      <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>
  <xsd:complexType name="GaugeMemberType">
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="Group" type="GroupType" minOccurs="1" />
      <xsd:element name="SortExpressions" type="SortExpressionsType"
        minOccurs="0" />
      <xsd:element name="GaugeMember" type="GaugeMemberType" minOccurs="0" />
      <xsd:any namespace="##other" processContents="skip" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>
  <xsd:complexType name="GaugeInputValueType">

```

```

<xsd:choice maxOccurs="unbounded">
  <xsd:element name="Value" type="xsd:string" minOccurs="1" />
  <xsd:element name="Formula" type="xsd:string" minOccurs="0" />
  <xsd:element name="MinPercent" type="xsd:string" minOccurs="0" />
  <xsd:element name="MaxPercent" type="xsd:string" minOccurs="0" />
  <xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
  <xsd:element name="AddConstant" type="xsd:string" minOccurs="0" />
  <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
  <xsd:element name="DataElementOutput" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Output" />
        <xsd:enumeration value="NoOutput" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="RadialGaugeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeTypeStart-->
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
    <xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="ClipContent" type="xsd:string" minOccurs="0" />
    <xsd:element name="AspectRatio" type="xsd:string" minOccurs="0" />
    <!--GaugeTypeEnd-->
    <xsd:element name="GaugeScales" type="RadialScalesType" minOccurs="0" />
    <xsd:element name="PivotX" type="xsd:string" minOccurs="0" />
    <xsd:element name="PivotY" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="LinearGaugeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeTypeStart-->
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
    <xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="ClipContent" type="xsd:string" minOccurs="0" />
    <xsd:element name="AspectRatio" type="xsd:string" minOccurs="0" />
    <!--GaugeTypeEnd-->
    <xsd:element name="GaugeScales" type="LinearScalesType" minOccurs="0" />
    <xsd:element name="Orientation" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="NumericIndicatorType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType"
      minOccurs="1" />
    <xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="NumericIndicatorRanges" type="NumericIndicatorRangesType"
      minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="DecimalDigitColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="DecimalDigits" type="xsd:string" minOccurs="0" />
    <xsd:element name="DigitColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Digits" type="xsd:string" minOccurs="0" />
    <xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="LedDimColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffString" type="xsd:string" minOccurs="0" />
    <xsd:element name="OutOfRangeString" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeparatorWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowDecimalPoint" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowLeadingZeros" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowSign" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="StateIndicatorType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType"
      minOccurs="1" />
    <xsd:element name="IndicatorStates" type="IndicatorStatesType"
      minOccurs="0" />
    <xsd:element name="StateImage" type="StateImageType" minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
    <xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="0" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>

```

```

    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="GaugeImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="GaugeLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextShadowOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="RadialScaleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeScaleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="ScaleRanges" type="ScaleRangesType" minOccurs="0" />
    <xsd:element name="ScaleLabels" type="ScaleLabelsType" minOccurs="0" />
    <xsd:element name="GaugeMajorTickMarks" type="GaugeTickMarksType"
      minOccurs="0" />
    <xsd:element name="GaugeMinorTickMarks" type="GaugeTickMarksType"
      minOccurs="0" />
    <xsd:element name="CustomLabels" type="CustomLabelsType" minOccurs="0" />
  </xsd:choice>

```



```

<xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
<xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
<xsd:element name="MaximumPin" type="ScalePinType" minOccurs="0" />
<xsd:element name="MinimumPin" type="ScalePinType" minOccurs="0" />
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
<xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
<xsd:element name="Logarithmic" type="xsd:string" minOccurs="0" />
<xsd:element name="LogarithmicBase" type="xsd:string" minOccurs="0" />
<xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
<xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
<xsd:element name="TickMarksOnTop" type="xsd:string" minOccurs="0" />
<!--GaugeScaleTypeEnd-->
<xsd:element name="GaugePointers" type="RadialPointersType" minOccurs="0" />
<xsd:element name="Radius" type="xsd:string" minOccurs="0" />
<xsd:element name="StartAngle" type="xsd:string" minOccurs="0" />
<xsd:element name="SweepAngle" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="LinearScaleType">
<xsd:choice minOccurs="0" maxOccurs="unbounded">
<!--GaugeScaleTypeStart-->
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="ScaleRanges" type="ScaleRangesType" minOccurs="0" />
<xsd:element name="ScaleLabels" type="ScaleLabelsType" minOccurs="0" />
<xsd:element name="GaugeMajorTickMarks" type="GaugeTickMarksType"
minOccurs="0" />
<xsd:element name="GaugeMinorTickMarks" type="GaugeTickMarksType"
minOccurs="0" />
<xsd:element name="CustomLabels" type="CustomLabelsType" minOccurs="0" />
<xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
<xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
<xsd:element name="MaximumPin" type="ScalePinType" minOccurs="0" />
<xsd:element name="MinimumPin" type="ScalePinType" minOccurs="0" />
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
<xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
<xsd:element name="Logarithmic" type="xsd:string" minOccurs="0" />
<xsd:element name="LogarithmicBase" type="xsd:string" minOccurs="0" />
<xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
<xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
<xsd:element name="TickMarksOnTop" type="xsd:string" minOccurs="0" />
<!--GaugeScaleTypeEnd-->
<xsd:element name="GaugePointers" type="LinearPointersType" minOccurs="0" />
<xsd:element name="StartMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="EndMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="RadialPointerType">
<xsd:choice minOccurs="0" maxOccurs="unbounded">
<!--GaugePointerTypeStart-->
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="GaugeInputValue" type="GaugeInputValueType"
minOccurs="0" />
<xsd:element name="PointerImage" type="PointerImageType" minOccurs="0" />
<xsd:element name="BarStart" type="xsd:string" minOccurs="0" />

```

```

<xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
<xsd:element name="MarkerLength" type="xsd:string" minOccurs="0" />
<xsd:element name="MarkerStyle" type="xsd:string" minOccurs="0" />
<xsd:element name="Placement" type="xsd:string" minOccurs="0" />
<xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
<xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
<xsd:element name="Type" type="xsd:string" minOccurs="0" />
<!--GaugePointerTypeEnd-->
<xsd:element name="PointerCap" type="PointerType" minOccurs="0" />
<xsd:element name="NeedleStyle" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="LinearPointerType">
<xsd:choice minOccurs="0" maxOccurs="unbounded">
<!--GaugePointerTypeStart-->
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="0" />
<xsd:element name="PointerImage" type="PointerType" minOccurs="0" />
<xsd:element name="BarStart" type="xsd:string" minOccurs="0" />
<xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
<xsd:element name="MarkerLength" type="xsd:string" minOccurs="0" />
<xsd:element name="MarkerStyle" type="xsd:string" minOccurs="0" />
<xsd:element name="Placement" type="xsd:string" minOccurs="0" />
<xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
<xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
<xsd:element name="Type" type="xsd:string" minOccurs="0" />
<!--GaugePointerTypeEnd-->
<xsd:element name="Thermometer" type="ThermometerType" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ThermometerType">
<xsd:choice minOccurs="0" maxOccurs="unbounded">
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="BulbOffset" type="xsd:string" minOccurs="0" />
<xsd:element name="BulbSize" type="xsd:string" minOccurs="0" />
<xsd:element name="ThermometerStyle" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="PointerCapType">
<xsd:choice minOccurs="0" maxOccurs="unbounded">
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="CapImage" type="CapImageType" minOccurs="0" />
<xsd:element name="OnTop" type="xsd:string" minOccurs="0" />
<xsd:element name="Reflection" type="xsd:string" minOccurs="0" />
<xsd:element name="CapStyle" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="skip" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="NumericIndicatorRangeType">
<xsd:choice minOccurs="0" maxOccurs="unbounded">
<xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
<xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />

```

```

    <xsd:element name="DecimalDigitColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="DigitColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="IndicatorStateType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="StateImage" type="StateImageType" minOccurs="0" />
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ScaleRangeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="BackgroundGradientType" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="StartWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="InRangeBarPointerColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="InRangeLabelColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="InRangeTickMarksColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ScaleLabelsType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="RotateLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowEndLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CustomLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkStyle" type="TickMarkStyleType" minOccurs="0" />
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="RotateLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TickMarkStyleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
    <xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shape" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="GaugeTickMarksType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--TickMarkStyleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
    <xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shape" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <!--TickMarkStyleTypeEnd-->
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ScalePinType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--TickMarkStyleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
    <xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shape" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <!--TickMarkStyleTypeEnd-->
    <xsd:element name="Location" type="xsd:string" minOccurs="0" />
    <xsd:element name="Enable" type="xsd:string" minOccurs="0" />
    <xsd:element name="PinLabel" type="PinLabelType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="PinLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="RotateLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="TopImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="StateImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="PointerImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="SizeType" minOccurs="0" />
    <xsd:element name="OffsetY" type="SizeType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CapImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="SizeType" minOccurs="0" />
    <xsd:element name="OffsetY" type="SizeType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="FrameImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->

```

```

    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <xsd:element name="ClipImage" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="BackFrameType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="FrameBackground" type="FrameBackgroundType"
minOccurs="0" />
    <xsd:element name="FrameImage" type="FrameImageType" minOccurs="0" />
    <xsd:element name="FrameStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="FrameShape" type="xsd:string" minOccurs="0" />
    <xsd:element name="FrameWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="GlassEffect" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="FrameBackgroundType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="RadialGaugesType">
  <xsd:sequence>
    <xsd:element name="RadialGauge" type="RadialGaugeType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="LinearGaugesType">
  <xsd:sequence>
    <xsd:element name="LinearGauge" type="LinearGaugeType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="NumericIndicatorsType">
  <xsd:sequence>
    <xsd:element name="NumericIndicator" type="NumericIndicatorType"
minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="StateIndicatorsType">
  <xsd:sequence>
    <xsd:element name="StateIndicator" type="StateIndicatorType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="GaugeImagesType">
  <xsd:sequence>
    <xsd:element name="GaugeImage" type="GaugeImageType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="GaugeLabelsType">
  <xsd:sequence>
    <xsd:element name="GaugeLabel" type="GaugeLabelType" minOccurs="1"

```

```

        maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="RadialScalesType">
    <xsd:sequence>
        <xsd:element name="RadialScale" type="RadialScaleType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="LinearScalesType">
    <xsd:sequence>
        <xsd:element name="LinearScale" type="LinearScaleType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="NumericIndicatorRangesType">
    <xsd:sequence>
        <xsd:element name="NumericIndicatorRange" type="NumericIndicatorRangeType"
            minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="IndicatorStatesType">
    <xsd:sequence>
        <xsd:element name="IndicatorState" type="IndicatorStateType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="RadialPointersType">
    <xsd:sequence>
        <xsd:element name="RadialPointer" type="RadialPointerType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="LinearPointersType">
    <xsd:sequence>
        <xsd:element name="LinearPointer" type="LinearPointerType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="ScaleRangesType">
    <xsd:sequence>
        <xsd:element name="ScaleRange" type="ScaleRangeType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
<xsd:complexType name="CustomLabelsType">
    <xsd:sequence>
        <xsd:element name="CustomLabel" type="CustomLabelType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
</xsd:schema>

```

## 5.4 RDL XML Schema for Version 2010/01

```

<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (c) Microsoft. All rights reserved. -->

```

```
<xsd:schema
targetNamespace="http://schemas.microsoft.com/sqlserver/reporting/2010/01/reportdefinition"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2010/01/reportdefinition"
elementFormDefault="qualified">
```

```
<xsd:annotation>
  <xsd:documentation>
```

The following schema describes the structure of the  
Report Definition Language (RDL) for Microsoft SQL Server 2008 R2.

THE SCHEMA IS PROVIDED TO YOU ON AN "AS IS" BASIS, AND MICROSOFT  
DISCLAIMS ALL WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING,  
WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS  
FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT, AS TO THE SCHEMA OR ANY  
PRODUCT OR OTHER ITEM THAT MAY BE DEVELOPED USING THE SCHEMA.

Without limiting the generality of the foregoing, Microsoft makes no  
warranty that any product or other item that may be developed using the  
schema, or any portion of the schema, will not infringe any copyright,  
patent, trade secret or other intellectual property right of any  
individual or legal entity in any country. It is your responsibility to  
obtain licenses to use any such intellectual property rights as appropriate.

MICROSOFT IS NOT LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN  
CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING, WITHOUT LIMITATION, ANY  
DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING LOST REVENUES OR LOST  
PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN  
ADVISED OF SUCH DAMAGES.

(c) Microsoft Corporation. All rights reserved.

```
</xsd:documentation>
</xsd:annotation>
<xsd:element name="Report">
  <xsd:complexType>
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="Description" type="StringLocIDType" minOccurs="0" />
      <xsd:element name="Author" type="xsd:string" minOccurs="0" />
      <xsd:element name="AutoRefresh" type="xsd:string" minOccurs="0" />
      <xsd:element name="InitialPageName" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataSources" type="DataSourcesType" minOccurs="0" />
      <xsd:element name="DataSets" type="DataSetsType" minOccurs="0" />
      <xsd:element name="ReportParameters" type="ReportParametersType" minOccurs="0" />
      <xsd:element name="Code" type="xsd:string" minOccurs="0" />
      <xsd:element name="EmbeddedImages" type="EmbeddedImagesType" minOccurs="0" />
      <xsd:element name="Language" type="xsd:string" minOccurs="0" />
      <xsd:element name="CodeModules" type="CodeModulesType" minOccurs="0" />
      <xsd:element name="Classes" type="ClassesType" minOccurs="0" />
      <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
      <xsd:element name="Variables" type="VariablesType" minOccurs="0" />
      <xsd:element name="DeferVariableEvaluation" type="xsd:boolean" minOccurs="0" />
      <xsd:element name="ConsumeContainerWhitespace" type="xsd:boolean"
        minOccurs="0" />
      <xsd:element name="DataTransform" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataSchema" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataElementStyle" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Attribute" />
            <xsd:enumeration value="Element" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:element name="ReportSections" type="ReportSectionsType" minOccurs="1" />
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```



```

        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
</xsd:element>
<xsd:complexType name="ReportSectionsType">
    <xsd:sequence>
        <xsd:element name="ReportSection" type="ReportSectionType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ReportSectionType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Body" type="BodyType" minOccurs="1" />
        <xsd:element name="Width" type="SizeType" minOccurs="1" />
        <xsd:element name="Page" type="PageType" minOccurs="1" />
        <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
        <xsd:element name="DataElementOutput" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Output" />
                    <xsd:enumeration value="NoOutput" />
                    <xsd:enumeration value="ContentsOnly" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ReportParametersType">
    <xsd:sequence>
        <xsd:element name="ReportParameter" type="ReportParameterType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ReportParameterType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="DataType">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Boolean" />
                    <xsd:enumeration value="DateTime" />
                    <xsd:enumeration value="Integer" />
                    <xsd:enumeration value="Float" />
                    <xsd:enumeration value="String" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Nullable" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="DefaultValue" type="DefaultValueType" minOccurs="0" />
        <xsd:element name="AllowBlank" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="Prompt" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="ValidValues" type="ValidValuesType" minOccurs="0" />
        <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="MultiValue" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="UsedInQuery" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="False" />
                    <xsd:enumeration value="True" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:choice>
    <xsd:any namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ValidValuesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataSetReference" type="DataSetReferenceType" minOccurs="0" />
    <xsd:element name="ParameterValues" type="ParameterValuesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataSetReferenceType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" />
    <xsd:element name="ValueField" type="xsd:string" />
    <xsd:element name="LabelField" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ParameterValuesType">
  <xsd:sequence>
    <xsd:element name="ParameterValue" type="ParameterValueType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ParameterValueType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="StringLocIDType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DefaultValueType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataSetReference" type="DataSetReferenceType" minOccurs="0" />
    <xsd:element name="Values" type="ValuesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ValuesType">
  <xsd:sequence>
    <xsd:element name="Value" type="xsd:string" minOccurs="1" maxOccurs="unbounded"
      nillable="true" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataSetsType">
  <xsd:sequence>
    <xsd:element name="DataSet" type="DataSetType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataSetType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Fields" type="FieldsType" minOccurs="0" />
    <xsd:element name="Query" type="QueryType" minOccurs="0" />
    <xsd:element name="SharedDataSet" type="SharedDataSetType" minOccurs="0" />
    <xsd:element name="CaseSensitivity" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="True" />
          <xsd:enumeration value="False" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>

```

```

    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="Collation" type="xsd:string" minOccurs="0" />
  <xsd:element name="AccentSensitivity" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="True" />
        <xsd:enumeration value="False" />
        <xsd:enumeration value="Auto" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="KanatypeSensitivity" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="True" />
        <xsd:enumeration value="False" />
        <xsd:enumeration value="Auto" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="WidthSensitivity" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="True" />
        <xsd:enumeration value="False" />
        <xsd:enumeration value="Auto" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
  <xsd:element name="InterpretSubtotalsAsDetails" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="True" />
        <xsd:enumeration value="False" />
        <xsd:enumeration value="Auto" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="FieldsType">
  <xsd:sequence>
    <xsd:element name="Field" type="FieldType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="StringWithDataModelAttribute">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="DataType" use="optional">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Boolean" />
            <xsd:enumeration value="DateTime" />
            <xsd:enumeration value="Integer" />
            <xsd:enumeration value="Float" />
            <xsd:enumeration value="String" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

```

```

<xsd:complexType name="FieldType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataField" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="StringWithDataModelAttribute" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="QueryType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataSourceName" type="xsd:string" />
    <xsd:element name="CommandType" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Text" />
          <xsd:enumeration value="StoredProcedure" />
          <xsd:enumeration value="TableDirect" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="CommandText" type="xsd:string" />
    <xsd:element name="QueryParameters" type="QueryParametersType" minOccurs="0" />
    <xsd:element name="Timeout" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="SharedDataSetType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="SharedDataSetReference" type="xsd:string" />
    <xsd:element name="QueryParameters" type="QueryParametersType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataSourcesType">
  <xsd:sequence>
    <xsd:element name="DataSource" type="DataSourceType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataSourceType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Transaction" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ConnectionProperties" type="ConnectionPropertiesType"
      minOccurs="0" />
    <xsd:element name="DataSourceReference" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ConnectionPropertiesType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataProvider" type="xsd:string" />
    <xsd:element name="ConnectionString" type="xsd:string" />
    <xsd:element name="IntegratedSecurity" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Prompt" type="StringLocIDType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="QueryParametersType">
  <xsd:sequence>
    <xsd:element name="QueryParameter" type="QueryParameterType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />

```

```

</xsd:complexType>
<xsd:complexType name="QueryParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="StringWithDataModelAttribute" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="CodeModulesType">
  <xsd:sequence>
    <xsd:element name="CodeModule" type="xsd:string" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ClassesType">
  <xsd:sequence>
    <xsd:element name="Class" type="ClassType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ClassType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ClassName" type="xsd:string" />
    <xsd:element name="InstanceName" type="xsd:normalizedString" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="BodyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="1" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="PageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="PageHeader" type="PageSectionType" minOccurs="0" />
    <xsd:element name="PageFooter" type="PageSectionType" minOccurs="0" />
    <xsd:element name="PageHeight" type="SizeType" minOccurs="0" />
    <xsd:element name="PageWidth" type="SizeType" minOccurs="0" />
    <xsd:element name="InteractiveHeight" type="SizeType" minOccurs="0" />
    <xsd:element name="InteractiveWidth" type="SizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="RightMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="TopMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="BottomMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="Columns" type="xsd:int" minOccurs="0" />
    <xsd:element name="ColumnSpacing" type="SizeType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="PageSectionType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Height" type="SizeType" />
    <xsd:element name="PrintOnFirstPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PrintOnLastPage" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="PrintBetweenSections" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

<xsd:complexType name="EmbeddedImagesType">
  <xsd:sequence>
    <xsd:element name="EmbeddedImage" type="EmbeddedImageType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="EmbeddedImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="MIMETYPE" type="xsd:string" />
    <xsd:element name="ImageData" type="xsd:string" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ReportItemsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Line" type="LineType" />
    <xsd:element name="Rectangle" type="RectangleType" />
    <xsd:element name="Textbox" type="TextboxType" />
    <xsd:element name="Image" type="ImageType" />
    <xsd:element name="Subreport" type="SubreportType" />
    <xsd:element name="Chart" type="ChartType" />
    <xsd:element name="GaugePanel" type="GaugePanelType" />
    <xsd:element name="Map" type="MapType" />
    <xsd:element name="Tablix" type="TablixType" />
    <xsd:element name="CustomReportItem" type="CustomReportItemType" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ActionInfoType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Actions" type="ActionsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ActionsType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Action" type="ActionType" minOccurs="1" maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ActionType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hyperlink" type="xsd:string" minOccurs="0" />
    <xsd:element name="Drillthrough" type="DrillthroughType" minOccurs="0" />
    <xsd:element name="BookmarkLink" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DrillthroughType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportName" type="xsd:string" />
    <xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="VisibilityType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToggleItem" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />

```

```

</xsd:complexType>
<xsd:complexType name="LineType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="RectangleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
    <xsd:element name="PageName" type="xsd:string" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

<xsd:complexType name="TextboxType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="Paragraphs" type="ParagraphsType" minOccurs="1" />
    <xsd:element name="CanGrow" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CanShrink" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="HideDuplicates" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToggleImage" type="ToggleImageType" minOccurs="0" />
    <xsd:element name="UserSort" type="UserSortType" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementStyle" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Auto" />
          <xsd:enumeration value="Attribute" />
          <xsd:enumeration value="Element" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ParagraphsType">
  <xsd:sequence>
    <xsd:element name="Paragraph" type="ParagraphType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ParagraphType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TextRuns" type="TextRunsType" minOccurs="1" />
    <xsd:element name="LeftIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="HangingIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="ListStyle" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Bulleted" />
          <xsd:enumeration value="Numbered" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>

```



```

</xsd:element>
  <xsd:element name="ListLevel" type="xsd:unsignedInt" minOccurs="0" />
  <xsd:element name="SpaceBefore" type="xsd:string" minOccurs="0" />
  <xsd:element name="SpaceAfter" type="xsd:string" minOccurs="0" />
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TextRunsType">
  <xsd:sequence>
    <xsd:element name="TextRun" type="TextRunType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TextRunType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Value" type="LocIDStringWithDataAttribute" minOccurs="1" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkupType" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ToggleImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="InitialState" type="xsd:string" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="Source">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="External" />
          <xsd:enumeration value="Embedded" />
          <xsd:enumeration value="Database" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="Sizing" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="AutoSize" />
          <xsd:enumeration value="Fit" />
          <xsd:enumeration value="FitProportional" />
          <xsd:enumeration value="Clip" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

</xsd:element>
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="SubreportType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="ReportName" type="xsd:string" />
    <xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
    <xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
    <xsd:element name="MergeTransactions" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="CustomReportItemType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Type" type="xsd:string" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
<xsd:element name="AltReportItem" type="ReportItemsType" minOccurs="0" />
<xsd:element name="CustomData" type="CustomDataType" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="CustomDataType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
    <xsd:element name="DataColumnHierarchy" type="DataColumnHierarchyType"
      minOccurs="0" />
    <xsd:element name="DataRowHierarchy" type="DataRowHierarchyType" minOccurs="0" />
    <xsd:element name="DataRows" type="DataRowsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataColumnHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataMembers" type="DataMembersType" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataRowHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataMembers" type="DataMembersType" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataMembersType">
  <xsd:sequence>
    <xsd:element name="DataMember" type="DataMemberType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataMemberType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
    <xsd:element name="Subtotal" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="DataMembers" type="DataMembersType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataRowsType">
  <xsd:sequence>
    <xsd:element name="DataRow" type="DataRowType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

<xsd:complexType name="DataRowType">
  <xsd:sequence>
    <xsd:element name="DataCell" type="DataCellType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataCellType">
  <xsd:sequence>
    <xsd:element name="DataValue" type="DataValueType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ParametersType">
  <xsd:sequence>
    <xsd:element name="Parameter" type="ParameterType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="Omit" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GroupType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="GroupExpressions" type="GroupExpressionsType" minOccurs="0" />
    <xsd:element name="ReGroupExpressions" type="GroupExpressionsType" minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
    <xsd:element name="PageName" type="xsd:string" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="Parent" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Variables" type="VariablesType" minOccurs="0" />
    <xsd:element name="DomainScope" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="VariablesType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Variable" type="VariableType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="VariableType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="StringWithDataTypeAttribute" minOccurs="1"
      maxOccurs="1" />
    <xsd:element name="Writable" type="xsd:boolean" minOccurs="0" maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />

```

```

    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
  <xsd:complexType name="GroupExpressionsType">
    <xsd:sequence>
      <xsd:element name="GroupExpression" type="xsd:string" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
  <xsd:complexType name="SortExpressionsType">
    <xsd:sequence>
      <xsd:element name="SortExpression" type="SortExpressionType" minOccurs="1"
        maxOccurs="unbounded" />
      <xsd:any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
  <xsd:complexType name="SortExpressionType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="Value" type="xsd:string" minOccurs="1" />
      <xsd:element name="Direction" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Ascending" />
            <xsd:enumeration value="Descending" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:element>
      <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
  <xsd:complexType name="ChartType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="Style" type="StyleType" minOccurs="0" />
      <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
      <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
      <xsd:element name="Top" type="SizeType" minOccurs="0" />
      <xsd:element name="Left" type="SizeType" minOccurs="0" />
      <xsd:element name="Height" type="SizeType" minOccurs="0" />
      <xsd:element name="Width" type="SizeType" minOccurs="0" />
      <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
      <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
      <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
      <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
      <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
      <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
      <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
      <xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
      <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
      <xsd:element name="PageName" type="xsd:string" minOccurs="0" />
      <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
      <xsd:element name="ChartSeriesHierarchy" type="ChartHierarchyType" />
      <xsd:element name="ChartCategoryHierarchy" type="ChartHierarchyType" />
      <xsd:element name="ChartData" type="ChartDataTypes" minOccurs="0" />
      <xsd:element name="ChartAreas" type="ChartAreasType" minOccurs="0" />
      <xsd:element name="ChartLegends" type="ChartLegendsType" minOccurs="0" />
      <xsd:element name="ChartTitles" type="ChartTitlesType" minOccurs="0" />
      <xsd:element name="DynamicHeight" type="xsd:string" minOccurs="0" />
      <xsd:element name="DynamicWidth" type="xsd:string" minOccurs="0" />
      <xsd:element name="Palette" type="xsd:string" minOccurs="0" />
      <xsd:element name="ChartCustomPaletteColors" type="ChartCustomPaletteColorsType"
        minOccurs="0" />
      <xsd:element name="PaletteHatchBehavior" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataElementOutput" minOccurs="0">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">

```

```

        <xsd:enumeration value="Output" />
        <xsd:enumeration value="NoOutput" />
        <xsd:enumeration value="ContentsOnly" />
        <xsd:enumeration value="Auto" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="ChartBorderSkin" type="ChartBorderSkinType" minOccurs="0" />
<xsd:element name="ChartNoDataMessage" type="ChartTitleType" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartHierarchyType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="ChartMembers" type="ChartMembersType" minOccurs="1"
            maxOccurs="1" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartMembersType">
    <xsd:sequence minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="ChartMember" type="ChartMemberType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartMemberType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="Group" type="GroupType" minOccurs="0" maxOccurs="1" />
        <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0"
            maxOccurs="1" />
        <xsd:element name="ChartMembers" type="ChartMembersType" minOccurs="0"
            maxOccurs="1" />
        <xsd:element name="Label" type="StringLocIDType" minOccurs="1" maxOccurs="1" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0"
            maxOccurs="1" />
        <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
        <xsd:element name="DataElementOutput" minOccurs="0" maxOccurs="1">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Output" />
                    <xsd:enumeration value="NoOutput" />
                    <xsd:enumeration value="ContentsOnly" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartAreasType">
    <xsd:sequence>
        <xsd:element name="ChartArea" type="ChartAreaType" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartAreaType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
        <xsd:element name="ChartCategoryAxes" type="ChartCategoryAxesType" minOccurs="0" />
        <xsd:element name="ChartValueAxes" type="ChartValueAxesType" minOccurs="0" />
        <xsd:element name="ChartThreeDProperties" type="ChartThreeDPropertiesType"
            minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="AlignOrientation" type="xsd:string" minOccurs="0" />
    </xsd:choice>

```

```

    <xsd:element name="ChartAlignType" type="ChartAlignTypeType" minOccurs="0" />
    <xsd:element name="ChartElementPosition" type="ChartElementPositionType"
      minOccurs="0" />
    <xsd:element name="ChartInnerPlotPosition" type="ChartElementPositionType"
      minOccurs="0" />
    <xsd:element name="AlignWithChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="EquallySizedAxesFont" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartAlignTypeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="AxesView" type="xsd:string" minOccurs="0" />
    <xsd:element name="Cursor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="InnerPlotPosition" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartElementPositionType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartTitlesType">
  <xsd:sequence>
    <xsd:element name="ChartTitle" type="ChartTitleType" minOccurs="0"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartTitleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Caption" type="StringLocIDType" minOccurs="1" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockToChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartElementPosition" type="ChartElementPositionType"
      minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendsType">
  <xsd:sequence>
    <xsd:element name="ChartLegend" type="ChartLegendType" minOccurs="0"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />

```

```

<xsd:element name="Position" type="xsd:string" minOccurs="0" />
<xsd:element name="Layout" type="xsd:string" minOccurs="0" />
<xsd:element name="DockToChartArea" type="xsd:string" minOccurs="0" />
<xsd:element name="DockOutsideChartArea" type="xsd:string" minOccurs="0" />
<xsd:element name="ChartElementPosition" type="ChartElementPositionType"
minOccurs="0" />
<xsd:element name="ChartLegendTitle" type="ChartLegendTitleType" minOccurs="0" />
<xsd:element name="AutoFitTextDisabled" type="xsd:string" minOccurs="0" />
<xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
<xsd:element name="ChartLegendColumns" type="ChartLegendColumnsType"
minOccurs="0" />
<xsd:element name="HeaderSeparator" type="xsd:string" minOccurs="0" />
<xsd:element name="HeaderSeparatorColor" type="xsd:string" minOccurs="0" />
<xsd:element name="ColumnSeparator" type="xsd:string" minOccurs="0" />
<xsd:element name="ColumnSeparatorColor" type="xsd:string" minOccurs="0" />
<xsd:element name="ColumnSpacing" type="xsd:string" minOccurs="0" />
<xsd:element name="InterlacedRows" type="xsd:string" minOccurs="0" />
<xsd:element name="InterlacedRowsColor" type="xsd:string" minOccurs="0" />
<xsd:element name="EquallySpacedItems" type="xsd:string" minOccurs="0" />
<xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
<xsd:element name="MaxAutoSize" type="xsd:string" minOccurs="0" />
<xsd:element name="TextWrapThreshold" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendTitleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Caption" type="StringLocIDType" />
    <xsd:element name="TitleSeparator" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartCustomPaletteColorsType">
  <xsd:sequence>
    <xsd:element name="ChartCustomPaletteColor" type="xsd:string"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartBorderSkinType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartBorderSkinType" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="ChartLegendColumnsType">
  <xsd:sequence>
    <xsd:element name="ChartLegendColumn" type="ChartLegendColumnType"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendColumnType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ColumnType">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Text" />
          <xsd:enumeration value="SeriesSymbol" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
  </xsd:choice>

```



```

        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="MinimumWidth" type="xsd:string" minOccurs="0" />
        <xsd:element name="MaximumWidth" type="xsd:string" minOccurs="0" />
        <xsd:element name="SeriesSymbolWidth" type="xsd:string" minOccurs="0" />
        <xsd:element name="SeriesSymbolHeight" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendColumnHeaderType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Value" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendCustomItemsType">
    <xsd:sequence>
        <xsd:element name="ChartLegendCustomItem" type="ChartLegendCustomItemType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendCustomItemType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="ChartLegendCustomItemCells"
            type="ChartLegendCustomItemCellsType" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
        <xsd:element name="Separator" type="xsd:string" minOccurs="0" />
        <xsd:element name="SeparatorColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendCustomItemCellsType">
    <xsd:sequence>
        <xsd:element name="ChartLegendCustomItemCell" type="ChartLegendCustomItemCellType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendCustomItemCellType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="CellType" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Text" />
                    <xsd:enumeration value="SeriesSymbol" />
                    <xsd:enumeration value="Image" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Text" type="xsd:string" minOccurs="0" />
        <xsd:element name="CellSpan" type="xsd:unsignedInt" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="ImageHeight" type="xsd:string" minOccurs="0" />
        <xsd:element name="ImageWidth" type="xsd:string" minOccurs="0" />
        <xsd:element name="SymbolHeight" type="xsd:string" minOccurs="0" />
        <xsd:element name="SymbolWidth" type="xsd:string" minOccurs="0" />
        <xsd:element name="Alignment" type="xsd:string" minOccurs="0" />
    </xsd:choice>

```

```

    <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartMarkerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <xsd:element name="Size" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartCategoryAxesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartAxis" type="ChartAxisType" maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartValueAxesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartAxis" type="ChartAxisType" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartAxisType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Visible" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartAxisTitle" type="ChartAxisTitleType" minOccurs="0" />
    <xsd:element name="Margin" type="xsd:string" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:element name="VariableAutoInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelIntervalType" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelIntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelIntervalOffsetType" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartMajorGridLines" type="ChartGridLinesType" minOccurs="0" />
    <xsd:element name="ChartMinorGridLines" type="ChartGridLinesType" minOccurs="0" />
    <xsd:element name="ChartMajorTickMarks" type="ChartTickMarksType" minOccurs="0" />
    <xsd:element name="ChartMinorTickMarks" type="ChartTickMarksType" minOccurs="0" />
    <xsd:element name="MarksAlwaysAtPlotEdge" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reverse" type="xsd:string" minOccurs="0" />
    <xsd:element name="CrossAt" type="xsd:string" minOccurs="0" />
    <xsd:element name="Location" type="xsd:string" minOccurs="0" />
    <xsd:element name="Interlaced" type="xsd:string" minOccurs="0" />
    <xsd:element name="InterlacedColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartStripLines" type="ChartStripLinesType" minOccurs="0" />
    <xsd:element name="Arrows" type="xsd:string" minOccurs="0" />
    <xsd:element name="Scalar" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Minimum" type="xsd:string" minOccurs="0" />
    <xsd:element name="Maximum" type="xsd:string" minOccurs="0" />
    <xsd:element name="LogScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="LogBase" type="xsd:string" minOccurs="0" />
    <xsd:element name="HideLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
    <xsd:element name="PreventFontShrink" type="xsd:string" minOccurs="0" />
    <xsd:element name="PreventFontGrow" type="xsd:string" minOccurs="0" />
    <xsd:element name="PreventLabelOffset" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:element name="PreventWordWrap" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowLabelRotation" type="xsd:string" minOccurs="0" />
    <xsd:element name="IncludeZero" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelsAutoFitDisabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaxFontSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="HideEndLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartAxisScaleBreak" type="ChartAxisScaleBreakType"
        minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartAxisTitleType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="Caption" type="StringLocIDType" />
        <xsd:element name="Position" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartStripLinesType">
    <xsd:sequence>
        <xsd:element name="ChartStripLine" type="ChartStripLineType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartStripLineType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="Title" type="xsd:string" minOccurs="0" />
        <xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
        <xsd:element name="StripWidth" type="xsd:string" minOccurs="0" />
        <xsd:element name="StripWidthType" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartAxisScaleBreakType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
        <xsd:element name="BreakLineType" type="xsd:string" minOccurs="0" />
        <xsd:element name="CollapsibleSpaceThreshold" type="xsd:string" minOccurs="0" />
        <xsd:element name="MaxNumberOfBreaks" type="xsd:string" minOccurs="0" />
        <xsd:element name="Spacing" type="xsd:string" minOccurs="0" />
        <xsd:element name="IncludeZero" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartDataTypes">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="ChartSeriesCollection" type="ChartSeriesCollectionType"
            minOccurs="1" maxOccurs="1" />
        <xsd:element name="ChartDerivedSeriesCollection"
            type="ChartDerivedSeriesCollectionType" minOccurs="0" />
    </xsd:choice>

```

```

    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartSeriesCollectionType">
  <xsd:sequence maxOccurs="unbounded" minOccurs="1">
    <xsd:element name="ChartSeries" type="ChartSeriesType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartDerivedSeriesCollectionType">
  <xsd:sequence maxOccurs="unbounded" minOccurs="1">
    <xsd:element name="ChartDerivedSeries" type="ChartDerivedSeriesType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartSeriesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartDataPoints" type="ChartDataPointsType" minOccurs="0" />
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <xsd:element name="Subtype" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartEmptyPoints" type="ChartEmptyPointsType" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartItemInLegend" type="ChartItemInLegendType" minOccurs="0" />
    <xsd:element name="ChartAreaName" type="xsd:string" minOccurs="0" />
    <xsd:element name="ValueAxisName" type="xsd:string" minOccurs="0" />
    <xsd:element name="CategoryAxisName" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartSmartLabel" type="ChartSmartLabelType" minOccurs="0" />
    <xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
    <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartDerivedSeriesType">
  <xsd:choice minOccurs="3" maxOccurs="unbounded">
    <xsd:element name="ChartSeries" type="ChartSeriesType" minOccurs="1" />
    <xsd:element name="SourceChartSeriesName" type="xsd:string" minOccurs="1" />
    <xsd:element name="DerivedSeriesFormula" minOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="BollingerBands" />
          <xsd:enumeration value="MovingAverage" />
          <xsd:enumeration value="ExponentialMovingAverage" />
          <xsd:enumeration value="TriangularMovingAverage" />
          <xsd:enumeration value="WeightedMovingAverage" />
          <xsd:enumeration value="MACD" />
          <xsd:enumeration value="DetrendedPriceOscillator" />
          <xsd:enumeration value="Envelopes" />
          <xsd:enumeration value="Performance" />
          <xsd:enumeration value="RateOfChange" />
          <xsd:enumeration value="RelativeStrengthIndex" />
          <xsd:enumeration value="StandardDeviation" />
          <xsd:enumeration value="TRIX" />
          <xsd:enumeration value="Mean" />
          <xsd:enumeration value="Median" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="ChartFormulaParameters" type="ChartFormulaParametersType"
      minOccurs="0" />
  </xsd:choice>
  <xsd:any namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

</xsd:complexType>
<xsd:complexType name="ChartFormulaParametersType">
  <xsd:sequence>
    <xsd:element name="ChartFormulaParameter" type="ChartFormulaParameterType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartFormulaParameterType">
  <xsd:choice minOccurs="1">
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Source" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartEmptyPointsType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
    <xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
    <xsd:element name="AxisLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartItemInLegendType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartDataPointsType">
  <xsd:sequence>
    <xsd:element name="ChartDataPoint" type="ChartDataPointType"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartDataPointType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="ChartDataPointValues" type="ChartDataPointValuesType"
      minOccurs="0" />
    <xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
    <xsd:element name="AxisLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
    <xsd:element name="ChartItemInLegend" type="ChartItemInLegendType" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartDataPointValuesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="x" type="xsd:string" minOccurs="0" />
    <xsd:element name="y" type="xsd:string" minOccurs="0" />
    <xsd:element name="Size" type="xsd:string" minOccurs="0" />
    <xsd:element name="High" type="xsd:string" minOccurs="0" />
    <xsd:element name="Low" type="xsd:string" minOccurs="0" />
    <xsd:element name="Start" type="xsd:string" minOccurs="0" />
    <xsd:element name="End" type="xsd:string" minOccurs="0" />
    <xsd:element name="Mean" type="xsd:string" minOccurs="0" />
    <xsd:element name="Median" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataValueType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartDataLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Visible" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Label" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="UseValueAsLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="Rotation" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartSmartLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Disabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowOutsidePlotArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutBackColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineAnchor" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowOverlapped" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerOverlapping" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaxMovingDistance" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinMovingDistance" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartNoMoveDirections" type="ChartNoMoveDirectionsType"
      minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartNoMoveDirectionsType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Up" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Right" type="xsd:string" minOccurs="0" />
    <xsd:element name="Down" type="xsd:string" minOccurs="0" />
    <xsd:element name="UpLeft" type="xsd:string" minOccurs="0" />
    <xsd:element name="UpRight" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

        <xsd:element name="DownLeft" type="xsd:string" minOccurs="0" />
        <xsd:element name="DownRight" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartThreeDPropertiesType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
        <xsd:element name="ProjectionMode" type="xsd:string" minOccurs="0" />
        <xsd:element name="Rotation" type="xsd:string" minOccurs="0" />
        <xsd:element name="Inclination" type="xsd:string" minOccurs="0" />
        <xsd:element name="Perspective" type="xsd:string" minOccurs="0" />
        <xsd:element name="DepthRatio" type="xsd:string" minOccurs="0" />
        <xsd:element name="Shading" type="xsd:string" minOccurs="0" />
        <xsd:element name="GapDepth" type="xsd:string" minOccurs="0" />
        <xsd:element name="WallThickness" type="xsd:string" minOccurs="0" />
        <xsd:element name="Clustered" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartGridLinesType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartTickMarksType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
        <xsd:element name="Type" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="Length" type="xsd:string" minOccurs="0" />
        <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="StyleType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Border" type="BorderType" minOccurs="0" />
        <xsd:element name="TopBorder" type="BorderType" minOccurs="0" />
        <xsd:element name="BottomBorder" type="BorderType" minOccurs="0" />
        <xsd:element name="LeftBorder" type="BorderType" minOccurs="0" />
        <xsd:element name="RightBorder" type="BorderType" minOccurs="0" />
        <xsd:element name="BackgroundColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="BackgroundGradientType" type="xsd:string" minOccurs="0" />
        <xsd:element name="BackgroundGradientEndColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="BackgroundHatchType" type="xsd:string" minOccurs="0" />
        <xsd:element name="BackgroundImage" type="BackgroundImageType" minOccurs="0" />
        <xsd:element name="FontStyle" type="xsd:string" minOccurs="0" />
        <xsd:element name="FontFamily" type="xsd:string" minOccurs="0" />
        <xsd:element name="FontSize" type="xsd:string" minOccurs="0" />
        <xsd:element name="FontWeight" type="xsd:string" minOccurs="0" />
        <xsd:element name="Format" type="xsd:string" minOccurs="0" />
        <xsd:element name="TextDecoration" type="xsd:string" minOccurs="0" />
        <xsd:element name="TextAlign" type="xsd:string" minOccurs="0" />
        <xsd:element name="TextEffect" type="xsd:string" minOccurs="0" />
        <xsd:element name="VerticalAlign" type="xsd:string" minOccurs="0" />
    </xsd:choice>

```

```

<xsd:element name="Color" type="xsd:string" minOccurs="0" />
<xsd:element name="ShadowColor" type="xsd:string" minOccurs="0" />
<xsd:element name="ShadowOffset" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingLeft" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingRight" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingTop" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingBottom" type="xsd:string" minOccurs="0" />
<xsd:element name="LineHeight" type="xsd:string" minOccurs="0" />
<xsd:element name="Direction" type="xsd:string" minOccurs="0" />
<xsd:element name="WritingMode" type="xsd:string" minOccurs="0" />
<xsd:element name="Language" type="xsd:string" minOccurs="0" />
<xsd:element name="UnicodeBiDi" type="xsd:string" minOccurs="0" />
<xsd:element name="Calendar" type="xsd:string" minOccurs="0" />
<xsd:element name="NumeralLanguage" type="xsd:string" minOccurs="0" />
<xsd:element name="NumeralVariant" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="BorderStyle">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Color" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="BackgroundImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Source">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="External" />
          <xsd:enumeration value="Embedded" />
          <xsd:enumeration value="Database" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundRepeat" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="FiltersType">
  <xsd:sequence>
    <xsd:element name="Filter" type="FilterType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="FilterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="FilterExpression" type="xsd:string" />
    <xsd:element name="Operator">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Equal" />
          <xsd:enumeration value="Like" />
          <xsd:enumeration value="NotEqual" />
          <xsd:enumeration value="GreaterThan" />
          <xsd:enumeration value="GreaterThanOrEqual" />
          <xsd:enumeration value="LessThan" />
          <xsd:enumeration value="LessThanOrEqual" />
          <xsd:enumeration value="TopN" />
          <xsd:enumeration value="BottomN" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>

```



```

        <xsd:enumeration value="TopPercent" />
        <xsd:enumeration value="BottomPercent" />
        <xsd:enumeration value="In" />
        <xsd:enumeration value="Between" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="FilterValues" type="FilterValuesType" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="FilterValuesType">
    <xsd:sequence>
        <xsd:element name="FilterValue" type="StringWithDataAttribute"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="UserSortType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="SortExpression" type="xsd:string" />
        <xsd:element name="SortExpressionScope" type="xsd:string" minOccurs="0" />
        <xsd:element name="SortTarget" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:simpleType name="SizeType">
    <xsd:restriction base="xsd:normalizedString">
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="StringLocIDType">
    <xsd:simpleContent>
        <xsd:extension base="xsd:string">
            <xsd:anyAttribute namespace="##other" processContents="lax" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="LocIDStringWithDataAttribute">
    <xsd:simpleContent>
        <xsd:extension base="StringWithDataAttribute">
            <xsd:attribute name="EvaluationMode" type="EvaluationModeType" default="Auto" />
            <xsd:anyAttribute namespace="##other" processContents="lax" />
        </xsd:extension>
    </xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="EvaluationModeType">
    <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Auto" />
        <xsd:enumeration value="Constant" />
    </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="CustomPropertiesType">
    <xsd:sequence>
        <xsd:element name="CustomProperty" type="CustomPropertyType"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="CustomPropertyType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Name" type="xsd:string" />
        <xsd:element name="Value" type="xsd:string" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixType">

```

```

<xsd:choice minOccurs="1" maxOccurs="unbounded">
  <xsd:element name="TablixCorner" type="TablixCornerType" minOccurs="0" />
  <xsd:element name="TablixBody" type="TablixBodyType" minOccurs="0" />
  <xsd:element name="TablixColumnHierarchy" type="TablixHierarchyType"
    minOccurs="1" />
  <xsd:element name="TablixRowHierarchy" type="TablixHierarchyType" minOccurs="1" />
  <xsd:element name="LayoutDirection" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="LTR" />
        <xsd:enumeration value="RTL" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="GroupsBeforeRowHeaders" type="xsd:unsignedInt" minOccurs="0" />
  <xsd:element name="RepeatColumnHeaders" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="RepeatRowHeaders" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="FixedColumnHeaders" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="FixedRowHeaders" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Style" type="StyleType" minOccurs="0" />
  <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
  <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
  <xsd:element name="Top" type="SizeType" minOccurs="0" />
  <xsd:element name="Left" type="SizeType" minOccurs="0" />
  <xsd:element name="Height" type="SizeType" minOccurs="0" />
  <xsd:element name="Width" type="SizeType" minOccurs="0" />
  <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
  <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
  <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
  <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
  <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
  <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
  <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
  <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
  <xsd:element name="PageName" type="xsd:string" minOccurs="0" />
  <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
  <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
  <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
  <xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="DataElementOutput" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Output" />
        <xsd:enumeration value="NoOutput" />
        <xsd:enumeration value="ContentsOnly" />
        <xsd:enumeration value="Auto" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixBodyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixColumns" type="TablixColumnsType" minOccurs="1"
      maxOccurs="1" />
    <xsd:element name="TablixRows" type="TablixRowsType" minOccurs="1" maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixCornerType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixCornerRows" type="TablixCornerRowsType" minOccurs="1"
      maxOccurs="1" />
  </xsd:choice>

```

```

        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixCornerRowsType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="TablixCornerRow" type="TablixCornerRowType" minOccurs="1"
            maxOccurs="unbounded">
            </xsd:element>
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixCornerRowType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="TablixCornerCell" type="TablixCornerCellType" minOccurs="0"
            maxOccurs="unbounded">
            </xsd:element>
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixCornerCellType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="CellContents" type="CellContentsType" minOccurs="0"
            maxOccurs="1" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixHierarchyType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1"
            maxOccurs="1" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixMembersType">
    <xsd:sequence minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="TablixMember" type="TablixMemberType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixMemberType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Group" type="GroupType" minOccurs="0" maxOccurs="1" />
        <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0"
            maxOccurs="1" />
        <xsd:element name="TablixHeader" type="TablixHeaderType" minOccurs="0"
            maxOccurs="1" />
        <xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="0"
            maxOccurs="1" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0"
            maxOccurs="1" />
        <xsd:element name="FixedData" type="xsd:boolean" minOccurs="0" maxOccurs="1" />
        <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" maxOccurs="1" />
        <xsd:element name="HideIfNoRows" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="RepeatOnNewPage" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="KeepWithGroup" minOccurs="0" maxOccurs="1">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="None" />
                    <xsd:enumeration value="Before" />
                    <xsd:enumeration value="After" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:choice>
</xsd:complexType>

```

```

<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixHeaderType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Size" type="SizeType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="CellContents" type="CellContentsType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="CellContentsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="ColSpan" type="xsd:unsignedInt" minOccurs="0" maxOccurs="1">
    </xsd:element>
    <xsd:element name="RowSpan" type="xsd:unsignedInt" minOccurs="0" maxOccurs="1">
    </xsd:element>
    <xsd:element name="Line" type="LineType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="Rectangle" type="RectangleType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="Textbox" type="TextboxType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="Image" type="ImageType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="Subreport" type="SubreportType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="Chart" type="ChartType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="GaugePanel" type="GaugePanelType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="Map" type="MapType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="CustomReportItem" type="CustomReportItemType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="Tablix" type="TablixType" minOccurs="0" maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixColumnsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="TablixColumn" type="TablixColumnType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixColumnType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Width" type="SizeType" minOccurs="1" maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixRowsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="TablixRow" type="TablixRowType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

<xsd:complexType name="TablixRowType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Height" type="SizeType" minOccurs="1" maxOccurs="1" />
    <xsd:element name="TablixCells" type="TablixCellsType" minOccurs="1"
      maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixCellsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="TablixCell" type="TablixCellType" minOccurs="1"
      maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixCellType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="CellContents" type="CellContentsType" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0"
      maxOccurs="1" />
    <xsd:element name="DataElementOutput" minOccurs="0" maxOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="PageBreakType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Disabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="ResetPageNumber" type="xsd:string" minOccurs="0" />
    <xsd:element name="BreakLocation" minOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Start" />
          <xsd:enumeration value="End" />
          <xsd:enumeration value="StartAndEnd" />
          <xsd:enumeration value="Between" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GaugePanelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--DataRegionTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
<xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
<xsd:element name="PageName" type="xsd:string" minOccurs="0" />
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<!--DataRegionTypeEnd-->
<xsd:element name="AntiAliasing" type="xsd:string" minOccurs="0" />
<xsd:element name="TextAntiAliasingQuality" type="xsd:string" minOccurs="0" />
<xsd:element name="AutoLayout" type="xsd:string" minOccurs="0" />
<xsd:element name="ShadowIntensity" type="xsd:string" minOccurs="0" />
<xsd:element name="RadialGauges" type="RadialGaugesType" minOccurs="0" />
<xsd:element name="LinearGauges" type="LinearGaugesType" minOccurs="0" />
<xsd:element name="NumericIndicators" type="NumericIndicatorsType" minOccurs="0" />
<xsd:element name="StateIndicators" type="StateIndicatorsType" minOccurs="0" />
<xsd:element name="GaugeImages" type="GaugeImagesType" minOccurs="0" />
<xsd:element name="GaugeLabels" type="GaugeLabelsType" minOccurs="0" />
<xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
<xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
<xsd:element name="GaugeMember" type="GaugeMemberType" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GaugeMemberType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="1" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
    <xsd:element name="GaugeMember" type="GaugeMemberType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GaugeInputValueType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="Formula" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinPercent" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaxPercent" type="xsd:string" minOccurs="0" />
    <xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
    <xsd:element name="AddConstant" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

<xsd:complexType name="RadialGaugeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeTypeStart-->
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
    <xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="ClipContent" type="xsd:string" minOccurs="0" />
    <xsd:element name="AspectRatio" type="xsd:string" minOccurs="0" />
    <!--GaugeTypeEnd-->
    <xsd:element name="GaugeScales" type="RadialScalesType" minOccurs="0" />
    <xsd:element name="PivotX" type="xsd:string" minOccurs="0" />
    <xsd:element name="PivotY" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="LinearGaugeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeTypeStart-->
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
    <xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="ClipContent" type="xsd:string" minOccurs="0" />
    <xsd:element name="AspectRatio" type="xsd:string" minOccurs="0" />
    <!--GaugeTypeEnd-->
    <xsd:element name="GaugeScales" type="LinearScalesType" minOccurs="0" />
    <xsd:element name="Orientation" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="NumericIndicatorType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
<xsd:element name="NumericIndicatorRanges" type="NumericIndicatorRangesType"
minOccurs="0" />
<xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
<xsd:element name="DecimalDigitColor" type="xsd:string" minOccurs="0" />
<xsd:element name="DecimalDigits" type="xsd:string" minOccurs="0" />
<xsd:element name="DigitColor" type="xsd:string" minOccurs="0" />
<xsd:element name="Digits" type="xsd:string" minOccurs="0" />
<xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="0" />
<xsd:element name="LedDimColor" type="xsd:string" minOccurs="0" />
<xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
<xsd:element name="OffString" type="xsd:string" minOccurs="0" />
<xsd:element name="OutOfRangeString" type="xsd:string" minOccurs="0" />
<xsd:element name="SeparatorColor" type="xsd:string" minOccurs="0" />
<xsd:element name="SeparatorWidth" type="xsd:string" minOccurs="0" />
<xsd:element name="ShowDecimalPoint" type="xsd:string" minOccurs="0" />
<xsd:element name="ShowLeadingZeros" type="xsd:string" minOccurs="0" />
<xsd:element name="ShowSign" type="xsd:string" minOccurs="0" />
<xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
<xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
<xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="StateIndicatorType">
<xsd:choice minOccurs="0" maxOccurs="unbounded">
<!--GaugePanelItemTypeStart-->
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
<xsd:element name="Height" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
<!--GaugePanelItemTypeEnd-->
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="0" />
<xsd:element name="IndicatorImage" type="IndicatorImageType" minOccurs="0" />
<xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="0" />
<xsd:element name="TransformationType" type="xsd:string" minOccurs="0" />
<xsd:element name="TransformationScope" type="xsd:string" minOccurs="0" />
<xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
<xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
<xsd:element name="IndicatorStates" type="IndicatorStatesType" minOccurs="0" />
<xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
<xsd:element name="Angle" type="xsd:string" minOccurs="0" />
<xsd:element name="ScaleFactor" type="xsd:string" minOccurs="0" />
<xsd:element name="StateDataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="StateDataElementOutput" minOccurs="0">
<xsd:simpleType>
<xsd:restriction base="xsd:string">
<xsd:enumeration value="Output" />
<xsd:enumeration value="NoOutput" />
</xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="GaugeImageType">
<xsd:choice minOccurs="0" maxOccurs="unbounded">

```



```

<!--GaugePanelItemTypeStart-->
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
<xsd:element name="Height" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
<!--GaugePanelItemTypeEnd-->
<xsd:element name="Source" type="xsd:string" minOccurs="1" />
<xsd:element name="Value" type="xsd:string" minOccurs="1" />
<xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
<xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
<xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
<xsd:element name="Angle" type="xsd:string" minOccurs="0" />
<xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GaugeLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextShadowOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="RadialScaleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeScaleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="ScaleRanges" type="ScaleRangesType" minOccurs="0" />
    <xsd:element name="ScaleLabels" type="ScaleLabelsType" minOccurs="0" />
    <xsd:element name="GaugeMajorTickMarks" type="GaugeTickMarksType" minOccurs="0" />
    <xsd:element name="GaugeMinorTickMarks" type="GaugeTickMarksType" minOccurs="0" />
    <xsd:element name="CustomLabels" type="CustomLabelsType" minOccurs="0" />
    <xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MaximumPin" type="ScalePinType" minOccurs="0" />
    <xsd:element name="MinimumPin" type="ScalePinType" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="Logarithmic" type="xsd:string" minOccurs="0" />
    <xsd:element name="LogarithmicBase" type="xsd:string" minOccurs="0" />
    <xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="TickMarksOnTop" type="xsd:string" minOccurs="0" />
    <!--GaugeScaleTypeEnd-->
    <xsd:element name="GaugePointers" type="RadialPointersType" minOccurs="0" />
    <xsd:element name="Radius" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="SweepAngle" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="LinearScaleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeScaleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="ScaleRanges" type="ScaleRangesType" minOccurs="0" />
    <xsd:element name="ScaleLabels" type="ScaleLabelsType" minOccurs="0" />
    <xsd:element name="GaugeMajorTickMarks" type="GaugeTickMarksType" minOccurs="0" />
    <xsd:element name="GaugeMinorTickMarks" type="GaugeTickMarksType" minOccurs="0" />
    <xsd:element name="CustomLabels" type="CustomLabelsType" minOccurs="0" />
    <xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MaximumPin" type="ScalePinType" minOccurs="0" />
    <xsd:element name="MinimumPin" type="ScalePinType" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="Logarithmic" type="xsd:string" minOccurs="0" />
    <xsd:element name="LogarithmicBase" type="xsd:string" minOccurs="0" />
    <xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="TickMarksOnTop" type="xsd:string" minOccurs="0" />
    <!--GaugeScaleTypeEnd-->
    <xsd:element name="GaugePointers" type="LinearPointersType" minOccurs="0" />
    <xsd:element name="StartMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="RadialPointerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePointerTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="PointerImage" type="PointerImageType" minOccurs="0" />
    <xsd:element name="BarStart" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerLength" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <!--GaugePointerTypeEnd-->
    <xsd:element name="PointerCap" type="PointerCapType" minOccurs="0" />
    <xsd:element name="NeedleStyle" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>

```

```

    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="LinearPointerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePointerTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="PointerImage" type="PointerImageType" minOccurs="0" />
    <xsd:element name="BarStart" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerLength" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <!--GaugePointerTypeEnd-->
    <xsd:element name="Thermometer" type="ThermometerType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ThermometerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="BulbOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="BulbSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="ThermometerStyle" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="PointerCapType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="CapImage" type="CapImageType" minOccurs="0" />
    <xsd:element name="OnTop" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reflection" type="xsd:string" minOccurs="0" />
    <xsd:element name="CapStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="NumericIndicatorRangeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="DecimalDigitColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="DigitColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="IndicatorStateType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="Color" type="xsd:string" minOccurs="1" />
    <xsd:element name="ScaleFactor" type="xsd:string" minOccurs="1" />
    <xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="1" />
  </xsd:choice>

```

```

        <xsd:element name="IndicatorImage" type="IndicatorImageType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ScaleRangeType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="BackgroundGradientType" type="xsd:string" minOccurs="0" />
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
        <xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
        <xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
        <xsd:element name="StartWidth" type="xsd:string" minOccurs="0" />
        <xsd:element name="EndWidth" type="xsd:string" minOccurs="0" />
        <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
        <xsd:element name="InRangeBarPointerColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="InRangeLabelColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="InRangeTickMarksColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
        <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ScaleLabelsType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
        <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
        <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
        <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
        <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
        <xsd:element name="RotateLabels" type="xsd:string" minOccurs="0" />
        <xsd:element name="ShowEndLabels" type="xsd:string" minOccurs="0" />
        <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
        <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="CustomLabelType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="TickMarkStyle" type="TickMarkStyleType" minOccurs="0" />
        <xsd:element name="Text" type="xsd:string" minOccurs="0" />
        <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
        <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
        <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
        <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
        <xsd:element name="RotateLabel" type="xsd:string" minOccurs="0" />
        <xsd:element name="Value" type="xsd:string" minOccurs="0" />
        <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
        <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TickMarkStyleType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
        <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
        <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
        <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
    </xsd:choice>

```

```

    <xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shape" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GaugeTickMarksType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--TickMarkStyleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
    <xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shape" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <!--TickMarkStyleTypeEnd-->
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ScalePinType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--TickMarkStyleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
    <xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shape" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <!--TickMarkStyleTypeEnd-->
    <xsd:element name="Location" type="xsd:string" minOccurs="0" />
    <xsd:element name="Enable" type="xsd:string" minOccurs="0" />
    <xsd:element name="PinLabel" type="PinLabelType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="PinLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="RotateLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TopImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />

```

```

        <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
        <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
        <!--BaseGaugeImageTypeEnd-->
        <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="IndicatorImageType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <!--BaseGaugeImageTypeStart-->
        <xsd:element name="Source" type="xsd:string" minOccurs="1" />
        <xsd:element name="Value" type="xsd:string" minOccurs="1" />
        <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
        <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
        <!--BaseGaugeImageTypeEnd-->
        <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="PointerImageType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <!--BaseGaugeImageTypeStart-->
        <xsd:element name="Source" type="xsd:string" minOccurs="1" />
        <xsd:element name="Value" type="xsd:string" minOccurs="1" />
        <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
        <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
        <!--BaseGaugeImageTypeEnd-->
        <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
        <xsd:element name="OffsetX" type="SizeType" minOccurs="0" />
        <xsd:element name="OffsetY" type="SizeType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="CapImageType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <!--BaseGaugeImageTypeStart-->
        <xsd:element name="Source" type="xsd:string" minOccurs="1" />
        <xsd:element name="Value" type="xsd:string" minOccurs="1" />
        <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
        <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
        <!--BaseGaugeImageTypeEnd-->
        <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="OffsetX" type="SizeType" minOccurs="0" />
        <xsd:element name="OffsetY" type="SizeType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="FrameImageType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <!--BaseGaugeImageTypeStart-->
        <xsd:element name="Source" type="xsd:string" minOccurs="1" />
        <xsd:element name="Value" type="xsd:string" minOccurs="1" />
        <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
        <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
        <!--BaseGaugeImageTypeEnd-->
        <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
        <xsd:element name="ClipImage" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="BackFrameType">

```

```

<xsd:choice minOccurs="0" maxOccurs="unbounded">
  <xsd:element name="Style" type="StyleType" minOccurs="0" />
  <xsd:element name="FrameBackground" type="FrameBackgroundType" minOccurs="0" />
  <xsd:element name="FrameImage" type="FrameImageType" minOccurs="0" />
  <xsd:element name="FrameStyle" type="xsd:string" minOccurs="0" />
  <xsd:element name="FrameShape" type="xsd:string" minOccurs="0" />
  <xsd:element name="FrameWidth" type="xsd:string" minOccurs="0" />
  <xsd:element name="GlassEffect" type="xsd:string" minOccurs="0" />
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="FrameBackgroundType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="RadialGaugesType">
  <xsd:sequence>
    <xsd:element name="RadialGauge" type="RadialGaugeType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="LinearGaugesType">
  <xsd:sequence>
    <xsd:element name="LinearGauge" type="LinearGaugeType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="NumericIndicatorsType">
  <xsd:sequence>
    <xsd:element name="NumericIndicator" type="NumericIndicatorType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="StateIndicatorsType">
  <xsd:sequence>
    <xsd:element name="StateIndicator" type="StateIndicatorType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GaugeImagesType">
  <xsd:sequence>
    <xsd:element name="GaugeImage" type="GaugeImageType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GaugeLabelsType">
  <xsd:sequence>
    <xsd:element name="GaugeLabel" type="GaugeLabelType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="RadialScalesType">
  <xsd:sequence>
    <xsd:element name="RadialScale" type="RadialScaleType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="LinearScalesType">
  <xsd:sequence>
    <xsd:element name="LinearScale" type="LinearScaleType" minOccurs="1"

```

```

        maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="NumericIndicatorRangesType">
    <xsd:sequence>
        <xsd:element name="NumericIndicatorRange" type="NumericIndicatorRangeType"
            minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="IndicatorStatesType">
    <xsd:sequence>
        <xsd:element name="IndicatorState" type="IndicatorStateType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="RadialPointersType">
    <xsd:sequence>
        <xsd:element name="RadialPointer" type="RadialPointerType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="LinearPointersType">
    <xsd:sequence>
        <xsd:element name="LinearPointer" type="LinearPointerType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ScaleRangesType">
    <xsd:sequence>
        <xsd:element name="ScaleRange" type="ScaleRangeType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="CustomLabelsType">
    <xsd:sequence>
        <xsd:element name="CustomLabel" type="CustomLabelType" minOccurs="1"
            maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="MapType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <!--ReportItemTypeStart-->
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:element name="Top" type="SizeType" minOccurs="0" />
        <xsd:element name="Left" type="SizeType" minOccurs="0" />
        <xsd:element name="Height" type="SizeType" minOccurs="0" />
        <xsd:element name="Width" type="SizeType" minOccurs="0" />
        <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
        <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
        <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
        <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
        <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
        <xsd:element name="DataElementOutput" minOccurs="0" />
        <!--ReportItemTypeEnd-->
        <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
        <xsd:element name="PageName" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="TileLanguage" type="xsd:string" minOccurs="0" />
    </xsd:choice>
</xsd:complexType>

```



```

<xsd:element name="MapLayers" type="MapLayersType" minOccurs="0" />
<xsd:element name="MapDataRegions" type="MapDataRegionsType" minOccurs="0" />
<xsd:element name="MapViewport" type="MapViewportType" minOccurs="1" />
<xsd:element name="MapLegends" type="MapLegendsType" minOccurs="0" />
<xsd:element name="MapTitles" type="MapTitlesType" minOccurs="0" />
<xsd:element name="MapDistanceScale" type="MapDistanceScaleType" minOccurs="0" />
<xsd:element name="MapColorScale" type="MapColorScaleType" minOccurs="0" />
<xsd:element name="MapBorderSkin" type="MapBorderSkinType" minOccurs="0" />
<xsd:element name="AntiAliasing" type="xsd:string" minOccurs="0" />
<xsd:element name="TextAntiAliasingQuality" type="xsd:string" minOccurs="0" />
<xsd:element name="ShadowIntensity" type="xsd:string" minOccurs="0" />
<xsd:element name="MaximumSpatialElementCount" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="MaximumTotalPointCount" type="xsd:unsignedInt" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapDataRegionsType">
  <xsd:sequence>
    <xsd:element name="MapDataRegion" type="MapDataRegionType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLayersType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapTileLayer" type="MapTileLayerType" />
    <xsd:element name="MapPolygonLayer" type="MapPolygonLayerType" />
    <xsd:element name="MapPointLayer" type="MapPointLayerType" />
    <xsd:element name="MapLineLayer" type="MapLineLayerType" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLegendsType">
  <xsd:sequence>
    <xsd:element name="MapLegend" type="MapLegendType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapTitlesType">
  <xsd:sequence>
    <xsd:element name="MapTitle" type="MapTitleType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapDataRegionType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="MapMember" type="MapMemberType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapMemberType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="1" />
    <xsd:element name="MapMember" type="MapMemberType" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapTileLayerType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapLayerStart-->
    <xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <!--MapLayerEnd-->
    <xsd:element name="TileStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseSecureConnection" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapTiles" type="MapTilesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapTilesType">
  <xsd:sequence>
    <xsd:element name="MapTile" type="MapTileType" minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapTileType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" minOccurs="1" />
    <xsd:element name="TileData" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPolygonLayerType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapLayerStart-->
    <xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <!--MapLayerEnd-->
    <!--MapVectorLayerStart-->
    <xsd:element name="MapDataRegionName" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType"
      minOccurs="0" />
    <xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType"
      minOccurs="0" />
    <xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
    <xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType" minOccurs="0" />
    <xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType"
      minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapVectorLayerEnd-->
    <xsd:element name="MapPolygonTemplate" type="MapPolygonTemplateType" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:element name="MapPolygonRules" type="MapPolygonRulesType" minOccurs="0" />
    <xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType" minOccurs="0" />
    <xsd:element name="MapCenterPointRules" type="MapPointRulesType" minOccurs="0" />
    <xsd:element name="MapPolygons" type="MapPolygonsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPointLayerType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapLayerStart-->
    <xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <!--MapLayerEnd-->
    <!--MapVectorLayerStart-->
    <xsd:element name="MapDataRegionName" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType"
      minOccurs="0" />
    <xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType" minOccurs="0" />
    <xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
    <xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType" minOccurs="0" />
    <xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType"
      minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapVectorLayerEnd-->
    <xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType" minOccurs="0" />
    <xsd:element name="MapPointRules" type="MapPointRulesType" minOccurs="0" />
    <xsd:element name="MapPoints" type="MapPointsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLineLayerType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapLayerStart-->
    <xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <!--MapLayerEnd-->
    <!--MapVectorLayerStart-->
    <xsd:element name="MapDataRegionName" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType"
      minOccurs="0" />
    <xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType"
      minOccurs="0" />
    <xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
    <xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType" minOccurs="0" />
    <xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType"
      minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

        <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<!--MapVectorLayerEnd-->
<xsd:element name="MapLineTemplate" type="MapLineTemplateType" minOccurs="0" />
<xsd:element name="MapLineRules" type="MapLineRulesType" minOccurs="0" />
<xsd:element name="MapLines" type="MapLinesType" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapShapefileType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="Source" type="xsd:string" minOccurs="1" />
        <xsd:element name="MapFieldNames" type="MapFieldNamesType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapSpatialDataSetType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="DataSetName" type="xsd:string" minOccurs="1" />
        <xsd:element name="SpatialField" type="xsd:string" minOccurs="1" />
        <xsd:element name="MapFieldNames" type="MapFieldNamesType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapSpatialDataRegionType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPolygonsType">
    <xsd:sequence>
        <xsd:element name="MapPolygon" type="MapPolygonType" minOccurs="1"
maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPointsType">
    <xsd:sequence>
        <xsd:element name="MapPoint" type="MapPointType" minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLinesType">
    <xsd:sequence>
        <xsd:element name="MapLine" type="MapLineType" minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPolygonType">
    <xsd:choice maxOccurs="unbounded">
        <!--MapSpatialElement Start-->
        <xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
        <xsd:element name="MapFields" type="MapFieldsType" minOccurs="0" />
        <!--MapSpatialElement End-->
    </xsd:choice>

```

```

    <xsd:element name="UseCustomPolygonTemplate" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapPolygonTemplate" type="MapPolygonTemplateType" minOccurs="0" />
    <xsd:element name="UseCustomCenterPointTemplate" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPointType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSpatialElement Start-->
    <xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
    <xsd:element name="MapFields" type="MapFieldsType" minOccurs="0" />
    <!--MapSpatialElement End-->
    <xsd:element name="UseCustomPointTemplate" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLineType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSpatialElement Start-->
    <xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
    <xsd:element name="MapFields" type="MapFieldsType" minOccurs="0" />
    <!--MapSpatialElement End-->
    <xsd:element name="UseCustomLineTemplate" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapLineTemplate" type="MapLineTemplateType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapFieldNamesType">
  <xsd:sequence>
    <xsd:element name="MapFieldName" type="xsd:string" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapFieldDefinitionsType">
  <xsd:sequence>
    <xsd:element name="MapFieldDefinition" type="MapFieldDefinitionType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapFieldDefinitionType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" minOccurs="1" />
    <xsd:element name="DataType" minOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Boolean" />
          <xsd:enumeration value="DateTime" />
          <xsd:enumeration value="Integer" />
          <xsd:enumeration value="Float" />
          <xsd:enumeration value="String" />
          <xsd:enumeration value="Decimal" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
  <xsd:any namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

</xsd:complexType>

<xsd:complexType name="MapFieldsType">
  <xsd:sequence>
    <xsd:element name="MapField" type="MapFieldType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapFieldType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapBindingFieldPairsType">
  <xsd:sequence>
    <xsd:element name="MapBindingFieldPair" type="MapBindingFieldPairType"
      minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapBindingFieldPairType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="FieldName" type="xsd:string" minOccurs="1" />
    <xsd:element name="BindingExpression" type="xsd:string" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPolygonTemplateType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSpatialElementTemplateType Start-->
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetY" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementLabel" type="xsd:string" minOccurs="0" />
    <!--MapSpatialElementTemplateType End-->
    <xsd:element name="ScaleFactor" type="xsd:string" minOccurs="0" />
    <xsd:element name="CenterPointOffsetX" type="xsd:string" minOccurs="0" />
    <xsd:element name="CenterPointOffsetY" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapMarkerTemplateType">
  <xsd:choice maxOccurs="unbounded">

```

```

<!--MapSpatialElementTemplateType Start-->
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="OffsetX" type="xsd:string" minOccurs="0" />
<xsd:element name="OffsetY" type="xsd:string" minOccurs="0" />
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="Label" type="xsd:string" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="DataElementLabel" type="xsd:string" minOccurs="0" />
<!--MapSpatialElementTemplateType End-->
<!--MapPointTemplateType Start-->
<xsd:element name="Size" type="xsd:string" minOccurs="0" />
<xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
<!--MapPointTemplateTypeType End-->
<xsd:element name="MapMarker" type="MapMarkerType" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLineTemplateType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSpatialElementTemplateType Start-->
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetY" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementLabel" type="xsd:string" minOccurs="0" />
    <!--MapSpatialElementTemplateType End-->
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapBucketsType">
  <xsd:sequence>
    <xsd:element name="MapBucket" type="MapBucketType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapBucketType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapColorRangeRuleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapAppearanceRule Start-->
    <xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapAppearanceRule End-->
    <!--MapColorRule Start-->
    <xsd:element name="ShowInColorScale" type="xsd:string" minOccurs="0" />
    <!--MapColorRule End-->
    <xsd:element name="StartColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="MiddleColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapColorPaletteRuleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapAppearanceRule Start-->
    <xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapAppearanceRule End-->
    <!--MapColorRule Start-->
    <xsd:element name="ShowInColorScale" type="xsd:string" minOccurs="0" />
    <!--MapColorRule End-->
    <xsd:element name="Palette" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapCustomColorRuleType">

```



```

<xsd:choice maxOccurs="unbounded">
  <!--MapAppearanceRule Start-->
  <xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
  <xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
  <xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
  <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
  <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
  <xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
  <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
  <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
  <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
  <xsd:element name="DataElementOutput" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Output" />
        <xsd:enumeration value="NoOutput" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <!--MapAppearanceRule End-->
  <!--MapColorRule Start-->
  <xsd:element name="ShowInColorScale" type="xsd:string" minOccurs="0" />
  <!--MapColorRule End-->
  <xsd:element name="MapCustomColors" type="MapCustomColorsType" minOccurs="1" />
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapCustomColorsType">
  <xsd:sequence>
    <xsd:element name="MapCustomColor" type="xsd:string" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapSizeRuleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapAppearanceRule Start-->
    <xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapAppearanceRule End-->
    <xsd:element name="StartSize" type="xsd:string" minOccurs="1" />
    <xsd:element name="EndSize" type="xsd:string" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapMarkerRuleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapAppearanceRule Start-->

```

```

<xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
<xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
<xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
<xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
<xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
<xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
<xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
<xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<!--MapAppearanceRule End-->
<xsd:element name="MapMarkers" type="MapMarkersType" minOccurs="1" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapMarkersType">
  <xsd:sequence>
    <xsd:element name="MapMarker" type="MapMarkerType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapMarkerType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapMarkerStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapMarkerImage" type="MapMarkerImageType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapMarkerImageType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Source" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPolygonRulesType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType" minOccurs="0" />
    <xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType" minOccurs="0" />
    <xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPointRulesType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType" minOccurs="0" />
    <xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType" minOccurs="0" />
    <xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType" minOccurs="0" />
    <xsd:element name="MapSizeRule" type="MapSizeRuleType" minOccurs="0" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

    <xsd:element name="MapMarkerRule" type="MapMarkerRuleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLineRulesType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType" minOccurs="0" />
    <xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType" minOccurs="0" />
    <xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType" minOccurs="0" />
    <xsd:element name="MapSizeRule" type="MapSizeRuleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapViewportType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSubItem Start-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
    <xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <!--MapSubItem End-->
    <xsd:element name="MapCoordinateSystem" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapProjection" type="xsd:string" minOccurs="0" />
    <xsd:element name="ProjectionCenterX" type="xsd:string" minOccurs="0" />
    <xsd:element name="ProjectionCenterY" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapCustomView" type="MapCustomViewType" minOccurs="0" />
    <xsd:element name="MapElementView" type="MapElementViewType" minOccurs="0" />
    <xsd:element name="MapDataBoundView" type="MapDataBoundViewType" minOccurs="0" />
    <xsd:element name="MapLimits" type="MapLimitsType" minOccurs="0" />
    <xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="SimplificationResolution" type="xsd:string" minOccurs="0" />
    <xsd:element name="ContentMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapMeridians" type="MapGridLinesType" minOccurs="0" />
    <xsd:element name="MapParallels" type="MapGridLinesType" minOccurs="0" />
    <xsd:element name="GridUnderContent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLimitsType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MinimumX" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumY" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumX" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumY" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapCustomViewType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Zoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="CenterX" type="xsd:string" minOccurs="0" />
    <xsd:element name="CenterY" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

<xsd:complexType name="MapElementViewType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Zoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="LayerName" type="xsd:string" minOccurs="1" />
    <xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType"
      minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapDataBoundViewType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Zoom" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapDistanceScaleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSubItem Start-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
    <xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <!--MapSubItem End-->
    <!--MapDockableSubItem Start-->
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <!--MapDockableSubItem End-->
    <xsd:element name="ScaleColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ScaleBorderColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapColorScaleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSubItem Start-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
    <xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <!--MapSubItem End-->
    <!--MapDockableSubItem Start-->
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <!--MapDockableSubItem End-->
    <xsd:element name="MapColorScaleTitle" type="MapColorScaleTitleType" minOccurs="0" />
    <xsd:element name="TickMarkLength" type="xsd:string" minOccurs="0" />
    <xsd:element name="ColorBarBorderColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelInterval" type="xsd:string" minOccurs="0" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

    <xsd:element name="LabelFormat" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelBehavior" type="xsd:string" minOccurs="0" />
    <xsd:element name="HideEndLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="RangeGapColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="NoDataText" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapColorScaleTitleType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Caption" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapTitleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSubItem Start-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
    <xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <!--MapSubItem End-->
    <!--MapDockableSubItem Start-->
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <!--MapDockableSubItem End-->
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextShadowOffset" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLegendType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSubItem Start-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
    <xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <!--MapSubItem End-->
    <!--MapDockableSubItem Start-->
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <!--MapDockableSubItem End-->
    <xsd:element name="Layout" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapLegendTitle" type="MapLegendTitleType" minOccurs="0" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

    <xsd:element name="AutoFitTextDisabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="InterlacedRows" type="xsd:string" minOccurs="0" />
    <xsd:element name="InterlacedRowsColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="EquallySpacedItems" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextWrapThreshold" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLegendTitleType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Caption" type="xsd:string" minOccurs="0" />
    <xsd:element name="TitleSeparator" type="xsd:string" minOccurs="0" />
    <xsd:element name="TitleSeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapGridLinesType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelPosition" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapBorderSkinType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapBorderSkinType" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLocationType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Unit" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapSizeType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Width" type="xsd:string" minOccurs="1" />
    <xsd:element name="Height" type="xsd:string" minOccurs="1" />
    <xsd:element name="Unit" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

</xsd:schema>

```

## 5.5 RDL XML Schema for Version 2011/01

**Note** RDL 2011/01 is not a complete schema. It is a micro-versioned schema and uses [RDL 2010/01](#) as its base schema. For more information about macro- and micro-versioned RDL schemas, see section [2.1](#).

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema
targetNamespace="http://schemas.microsoft.com/sqlserver/reporting/2011/01/reportdefinition"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns="http://schemas.microsoft.com/sqlserver/reporting/2011/01/reportdefinition"

xmlns:rdl2010="http://schemas.microsoft.com/sqlserver/reporting/2010/01/reportdefinition"
  elementFormDefault="qualified">
  <xsd:annotation>
    <xsd:documentation>

      The following schema describes the structure of the
      Report Definition Language (RDL) for Microsoft SQL Server 2008 R2.

      THE SCHEMA IS PROVIDED TO YOU ON AN "AS IS" BASIS, AND MICROSOFT
      DISCLAIMS ALL WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING,
      WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS
      FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT, AS TO THE SCHEMA OR ANY
      PRODUCT OR OTHER ITEM THAT MAY BE DEVELOPED USING THE SCHEMA.

      Without limiting the generality of the foregoing, Microsoft makes no
      warranty that any product or other item that may be developed using the
      schema, or any portion of the schema, will not infringe any copyright,
      patent, trade secret or other intellectual property right of any
      individual or legal entity in any country. It is your responsibility to
      obtain licenses to use any such intellectual property rights as appropriate.

      MICROSOFT IS NOT LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN
      CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING, WITHOUT LIMITATION, ANY
      DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING LOST REVENUES OR LOST
      PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN
      ADVISED OF SUCH DAMAGES.

      (c) Microsoft Corporation. All rights reserved.

    </xsd:documentation>
  </xsd:annotation>

  <xsd:import
namespace="http://schemas.microsoft.com/sqlserver/reporting/2010/01/reportdefinition"/>

  <!--The following element is an RDL property that indicates if a Tablix is scrollable.
Needs to be added under the Tablix element -->
  <xsd:element name ="CanScroll" type="xsd:boolean" />

  <!--The following element is an RDL property that indicates if a Textbox is vertically
scrollable.
Needs to be added under the Textbox element -->
  <xsd:element name ="CanScrollVertically" type="xsd:boolean" />

  <!-- May be placed on an Group to indicate the data is pre-grouped
by the query -->
  <xsd:element name="NaturalGroup" type="xsd:boolean" />

  <!--The following element is an RDL property that defines the dataset to use for this
scope.
Needs to be added under the following elements: Group, TablixCell, ChartDataPoint,
DataValue -->
  <xsd:element name ="DataSetName" type="xsd:string" />

  <!--The following element is an RDL property that defines the relationship to use for
correlating data in the parent scope with the data in the containing scope.
```

Needs to be added under the following elements: DataRegion, Group -->  
<xsd:element name="Relationship" type="RelationshipType" />

<!--The following element is an RDL property that defines the relationship to use for correlating data in the parent scope with the data in the containing scope. Needs to be added under the following elements: TablixCell, ChartDataPoint, DataValue -->  
<xsd:element name="Relationships" type="RelationshipsType" />

<!--The following element is an RDL property that represents default relationships to use between

the containing data set and other data sets when nested scopes use different data sets. Needs to be added under the DataSet element -->

<xsd:element name="DefaultRelationships" type="DefaultRelationshipsType" />

```
<xsd:complexType name="RelationshipsType">
  <xsd:sequence>
    <xsd:element name="Relationship" type="RelationshipType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

```
<xsd:complexType name="RelationshipType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ParentScope" type="xsd:string" minOccurs="0" />
    <xsd:element name="NaturalJoin" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="JoinConditions" type="JoinConditionsType" minOccurs="0" />
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

```
<xsd:complexType name="JoinConditionsType">
  <xsd:sequence>
    <xsd:element name="JoinCondition" type="JoinConditionType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

```
<xsd:complexType name="JoinConditionType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ForeignKey" type="xsd:string" />
    <xsd:element name="PrimaryKey" type="xsd:string" />
    <xsd:element name="SortDirection" minOccurs="0" maxOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Ascending"/>
          <xsd:enumeration value="Descending"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

```
<xsd:complexType name="DefaultRelationshipsType">
  <xsd:sequence>
    <xsd:element name="DefaultRelationship" type="DefaultRelationshipType" minOccurs="1"
      maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

```
<xsd:complexType name="DefaultRelationshipType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="RelatedDataSet" type="xsd:string" />
    <xsd:element name="NaturalJoin" type="xsd:boolean" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
```



```

        <xsd:element name="JoinConditions" type="JoinConditionsType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<!-- May be placed on a SortExpression element to indicate the data is pre-sorted
by the query and that sort condition -->
<xsd:element name="NaturalSort" type="xsd:boolean" />

<!-- May be placed on a SortExpression element to indicate the sort should be deferred. -->
<xsd:element name="DeferredSort" type="xsd:boolean" />

<!--The following element is an RDL property to represent a Tablix as a Band.
Needs to be added under the Tablix element -->
<xsd:element name="BandLayoutOptions">
    <xsd:complexType>
        <xsd:choice minOccurs="1" maxOccurs="unbounded">
            <xsd:element name="RowCount" minOccurs="0" maxOccurs="1">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:unsignedInt">
                        <xsd:minInclusive value="1"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="ColumnCount" minOccurs="0" maxOccurs="1">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:unsignedInt">
                        <xsd:minInclusive value="1"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:element>
            <xsd:element name="Coverflow" type="CoverflowType" minOccurs="0"/>
            <xsd:element name="PlayAxis" type="PlayAxisType" minOccurs="0"/>
            <xsd:element name="Tabstrip" type="TabstripType" minOccurs="0"/>
            <xsd:any namespace="##other" processContents="lax" />
        </xsd:choice>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
</xsd:element>

<xsd:complexType name="NavigationItemType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="ReportItemReference" type="xsd:string" minOccurs="0"
            maxOccurs="1"/>
        <xsd:element name="ReportItem" type="rdl2010:ReportItemsType" minOccurs="0"
            maxOccurs="1"/>
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="CoverflowType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="NavigationItem" type="NavigationItemType" minOccurs="0"
            maxOccurs="1"/>
        <xsd:element name="Slider" type="SliderType" minOccurs="0" maxOccurs="1"/>
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="TabstripType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="NavigationItem" type="NavigationItemType" minOccurs="0"
            maxOccurs="1"/>
        <xsd:element name="Slider" type="SliderType" minOccurs="0" maxOccurs="1"/>
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
</xsd:complexType>

```

```

    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="PlayAxisType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Slider" type="SliderType" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="DockingOption" minOccurs="0" maxOccurs="1">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Top"/>
          <xsd:enumeration value="Bottom"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="SliderType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="LabelData" type="LabelDataType" minOccurs="0" maxOccurs="1"/>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="LabelDataType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="Key" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="Label" type="xsd:string" minOccurs="0" maxOccurs="1"/>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<!--The following elements are RDL properties to represent navigation UI for banding.
Needs to be added under the Tablix element -->
<xsd:element name="LeftMargin" type="xsd:string" />
<xsd:element name="RightMargin" type="xsd:string" />
<xsd:element name="TopMargin" type="xsd:string" />
<xsd:element name="BottomMargin" type="xsd:string" />

<!-- Add the following under the ChartDataPointValuesType. This is for Chart Highlighting
-->
<xsd:element name="HighlightX" type="xsd:string" />
<xsd:element name="HighlightY" type="xsd:string" />
<xsd:element name="HighlightSize" type="xsd:string" />

<!--The following element is an RDL property that defines how to interpret subtotal data in
queries.
May be under the RDL Field element -->
<xsd:element name="AggregateIndicatorField" type="xsd:string" />

<!--The following element is an RDL property that defines how nulls should be sorted.
May be under the RDL DataSet element -->
<xsd:element name="NullsAsBlanks" type="xsd:boolean" />

<!--The following element is an RDL property that defines the culture name to use for
collation information.
May be under the RDL DataSet element -->
<xsd:element name="CollationCulture" type="xsd:string" />

<!--The following element is an RDL property that defines extra information, such as the key
of a database image used for async retrieval.
May be under the RDL Image element -->

```

```

<xsd:element name ="Tag" type="xsd:string" />

<!--The following attribute is an RDL property that defines a unqie name.
May be under the RDL ReportSection element -->
<xsd:attribute name="Name" type="xsd:string" />

</xsd:schema>

```

## 5.6 RDL XML Schema for Version 2012/01

**Note** RDL 2012/01 is not a complete schema. It is a micro-versioned schema and uses [RDL 2010/01](#) as its base schema. For more information about macro- and micro-versioned RDL schemas, see section [2.1](#).

```

<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (c) Microsoft. All rights reserved. -->
<xsd:schema
targetNamespace="http://schemas.microsoft.com/sqlserver/reporting/2012/01/reportdefinition"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2012/01/reportdefinition"
elementFormDefault="qualified">
<xsd:annotation>
<xsd:documentation>

The following schema describes the structure of the
Report Definition Language (RDL) for Microsoft SQL Server 2012.

THE SCHEMA IS PROVIDED TO YOU ON AN "AS IS" BASIS, AND MICROSOFT
DISCLAIMS ALL WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING,
WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS
FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT, AS TO THE SCHEMA OR ANY
PRODUCT OR OTHER ITEM THAT MAY BE DEVELOPED USING THE SCHEMA.

Without limiting the generality of the foregoing, Microsoft makes no
warranty that any product or other item that may be developed using the
schema, or any portion of the schema, will not infringe any copyright,
patent, trade secret or other intellectual property right of any
individual or legal entity in any country. It is your responsibility to
obtain licenses to use any such intellectual property rights as appropriate.

MICROSOFT IS NOT LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN
CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING, WITHOUT LIMITATION, ANY
DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING LOST REVENUES OR LOST
PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN
ADVISED OF SUCH DAMAGES.

(c) Microsoft Corporation. All rights reserved.

</xsd:documentation>
</xsd:annotation>

<!-- An RDL property describing additional subtypes for a chart.
ParentElements: rd12010:ChartSeries -->
<xsd:element name ="Subtype" type="xsd:string" />

<!-- Indicates how the report embeds the images.
ParentElements: rd12010:Image, rd12010:BackgroundImage -->
<xsd:element name ="EmbeddingMode">
<xsd:simpleType>
<xsd:restriction base="xsd:string">
<xsd:enumeration value="Inline"/>
<xsd:enumeration value="Package"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:element>

```

```

<!-- Indicates how the report section layout direction (RTL or LTR)
ParentElements: rd12010:ReportSection -->
<xsd:element name="LayoutDirection">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="LTR"/>
      <xsd:enumeration value="RTL"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>

<!-- Indicates if a value should be interpreted as a literal value or a reference to a
theme property for the value.-->
<xsd:attribute name="ValueType">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Constant" />
      <xsd:enumeration value="ThemeReference" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:attribute>

<!-- Base type for an element that uses the ValueType attribute -->
<xsd:complexType name="StringWithValueTypeAttribute">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute ref="ValueType" use="optional" />
      <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<!-- ValueType aware Style elements -->
<xsd:element name="FontFamily" type="StringWithValueTypeAttribute" />
<xsd:element name="Color" type="StringWithValueTypeAttribute" />
<xsd:element name="BackgroundColor" type="StringWithValueTypeAttribute" />

<!-- Indicates whether the TablixHierarchy or ChartCategoryHierarchy is in drilling mode
ParentElements: rd12010:TablixRowHierarchy, rd12010:TablixColumnHierarchy,
rd12010:ChartCategoryHierarchy -->
<xsd:element name="EnableDrilldown" type="xsd:boolean" />

<!-- Repeat options for background images -->
<xsd:element name="BackgroundRepeat" type="xsd:string" />

<!-- Transparency setting for background images -->
<xsd:element name="Transparency" type="xsd:string" />

</xsd:schema>

```

## 5.7 RDL XML Schema for Version 2013/01

**Note** RDL 2013/01 is not a complete schema. It is a micro-versioned schema and uses [RDL 2010/01](#) as its base schema. For more information about macro- and micro-versioned RDL schemas, see section [2.1](#).

```

<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (c) Microsoft. All rights reserved. -->
<xsd:schema
targetNamespace="http://schemas.microsoft.com/sqlserver/reporting/2013/01/reportdefinition"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2013/01/reportdefinition"
elementFormDefault="qualified">
  <xsd:annotation>
    <xsd:documentation>

```

The following schema describes the structure of the Report Definition Language (RDL) for Microsoft SQL Server 2012.

THE SCHEMA IS PROVIDED TO YOU ON AN "AS IS" BASIS, AND MICROSOFT DISCLAIMS ALL WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT, AS TO THE SCHEMA OR ANY PRODUCT OR OTHER ITEM THAT MAY BE DEVELOPED USING THE SCHEMA.

Without limiting the generality of the foregoing, Microsoft makes no warranty that any product or other item that may be developed using the schema, or any portion of the schema, will not infringe any copyright, patent, trade secret or other intellectual property right of any individual or legal entity in any country. It is your responsibility to obtain licenses to use any such intellectual property rights as appropriate.

MICROSOFT IS NOT LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING, WITHOUT LIMITATION, ANY DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING LOST REVENUES OR LOST PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN ADVISED OF SUCH DAMAGES.

(c) Microsoft Corporation. All rights reserved.

```
</xsd:documentation>
</xsd:annotation>

<!-- A collection of field references used as keys.  Supersedes rdl2011:LabelData.Key.
ParentElement: rdl2011:LabelData -->
<xsd:element name="KeyFields">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Key" type="xsd:string" minOccurs="1" maxOccurs="unbounded" />
      <xsd:any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
</xsd:element>

<!--A collection of values that define extra information, such as the key of a database
image used for async retrieval.
May be under the RDL Image element -->
<xsd:element name="Tags">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="Tag" type="xsd:string" minOccurs="1" maxOccurs="unbounded" />
      <xsd:any namespace="##other" processContents="lax" minOccurs="0"
        maxOccurs="unbounded"/>
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
</xsd:element>

<!-- Add the following under the ChartDataPointValuesType.  This is for Chart data point
value formatting -->
<xsd:element name="FormatX" type="xsd:string" />
<xsd:element name="FormatY" type="xsd:string" />
<xsd:element name="FormatSize" type="xsd:string" />
<xsd:element name="CurrencyLanguageX" type="xsd:string" />
<xsd:element name="CurrencyLanguageY" type="xsd:string" />
<xsd:element name="CurrencyLanguageSize" type="xsd:string" />

<!-- Add the following under the Style element for cell level formatting -->
<xsd:element name="CurrencyLanguage" type="xsd:string" />
</xsd:schema>
</xsd:schema>
```

## 5.8 RDL XML Schema for Version 2016/01

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (c) Microsoft. All rights reserved. -->
<xsd:schema
targetNamespaces="http://schemas.microsoft.com/sqlserver/reporting/2016/01/reportdefinition"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2016/01/reportdefinition"
elementFormDefault="qualified">

  <xsd:annotation>
    <xsd:documentation>
```

The following schema describes the structure of the  
Report Definition Language (RDL) for Microsoft SQL Server 2016.

THE SCHEMA IS PROVIDED TO YOU ON AN "AS IS" BASIS, AND MICROSOFT  
DISCLAIMS ALL WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING,  
WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS  
FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT, AS TO THE SCHEMA OR ANY  
PRODUCT OR OTHER ITEM THAT MAY BE DEVELOPED USING THE SCHEMA.

Without limiting the generality of the foregoing, Microsoft makes no  
warranty that any product or other item that may be developed using the  
schema, or any portion of the schema, will not infringe any copyright,  
patent, trade secret or other intellectual property right of any  
individual or legal entity in any country. It is your responsibility to  
obtain licenses to use any such intellectual property rights as appropriate.

MICROSOFT IS NOT LIABLE FOR ANY DAMAGES OF ANY KIND ARISING OUT OF OR IN  
CONNECTION WITH THE USE OF THE SCHEMA, INCLUDING, WITHOUT LIMITATION, ANY  
DIRECT, INDIRECT, INCIDENTAL, CONSEQUENTIAL (INCLUDING LOST REVENUES OR LOST  
PROFITS), PUNITIVE OR SPECIAL DAMAGES, WHETHER OR NOT MICROSOFT HAS BEEN  
ADVISED OF SUCH DAMAGES.

(c) Microsoft Corporation. All rights reserved.

```
</xsd:documentation>
</xsd:annotation>
<xsd:element name="Report">
  <xsd:complexType>
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="DefaultFontFamily" type="xsd:string" minOccurs="0"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2016/01/reportdefinition/defaultfontf
amily" />
      <xsd:element name="AuthoringMetadata" type="xsd:string" minOccurs="0"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2016/01/reportdefinition/authoringmet
adata" />
      <xsd:element name="Description" type="StringLocIDType" minOccurs="0" />
      <xsd:element name="Author" type="xsd:string" minOccurs="0" />
      <xsd:element name="AutoRefresh" type="xsd:string" minOccurs="0" />
      <xsd:element name="InitialPageName" type="xsd:string" minOccurs="0" />
      <xsd:element name="DataSources" type="DataSourcesType" minOccurs="0" />
      <xsd:element name="DataSets" type="DataSetsType" minOccurs="0" />
      <xsd:element name="ReportParameters" type="ReportParametersType" minOccurs="0" />
      <xsd:element name="ReportParametersLayout" type="ReportParametersLayoutType"
minOccurs="0" />
      <xsd:element name="Code" type="xsd:string" minOccurs="0" />
      <xsd:element name="EmbeddedImages" type="EmbeddedImagesType" minOccurs="0" />
      <xsd:element name="Language" type="xsd:string" minOccurs="0" />
      <xsd:element name="CodeModules" type="CodeModulesType" minOccurs="0" />
      <xsd:element name="Classes" type="ClassesType" minOccurs="0" />
      <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
      <xsd:element name="Variables" type="VariablesType" minOccurs="0" />
      <xsd:element name="DeferVariableEvaluation" type="xsd:boolean" minOccurs="0" />
      <xsd:element name="ConsumeContainerWhitespace" type="xsd:boolean" minOccurs="0"
/>
    />
  />
</xsd:element>
</xsd:complexType>
</xsd:element>
</xsd:schema>
```

```

    <xsd:element name="DataTransform" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataSchema" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementStyle" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Attribute" />
          <xsd:enumeration value="Element" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="ReportSections" type="ReportSectionsType" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="MustUnderstand" type="MustUnderstandType" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
</xsd:element>
<xsd:simpleType name="MustUnderstandType">
  <xsd:list itemType="xsd:token" />
</xsd:simpleType>
<xsd:complexType name="ReportSectionsType">
  <xsd:sequence>
    <xsd:element name="ReportSection" type="ReportSectionType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ReportSectionType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Body" type="BodyType" minOccurs="1" />
    <xsd:element name="Width" type="SizeType" minOccurs="1" />
    <xsd:element name="Page" type="PageType" minOccurs="1" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ReportParametersType">
  <xsd:sequence>
    <xsd:element name="ReportParameter" type="ReportParameterType" maxOccurs="unbounded"
/>
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ReportParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataType">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Boolean" />
          <xsd:enumeration value="DateTime" />
          <xsd:enumeration value="Integer" />
          <xsd:enumeration value="Float" />
          <xsd:enumeration value="String" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Nullable" type="xsd:boolean" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="DefaultValue" type="DefaultValueType" minOccurs="0" />
<xsd:element name="AllowBlank" type="xsd:boolean" minOccurs="0" />
<xsd:element name="Prompt" type="StringLocIDType" minOccurs="0" />
<xsd:element name="ValidValues" type="ValidValuesType" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" />
<xsd:element name="MultiValue" type="xsd:boolean" minOccurs="0" />
<xsd:element name="UsedInQuery" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="False" />
      <xsd:enumeration value="True" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ValidValuesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataSetReference" type="DataSetReferenceType" minOccurs="0" />
    <xsd:element name="ParameterValues" type="ParameterValuesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataSetReferenceType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" />
    <xsd:element name="ValueField" type="xsd:string" />
    <xsd:element name="LabelField" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ParameterValuesType">
  <xsd:sequence>
    <xsd:element name="ParameterValue" type="ParameterValueType" maxOccurs="unbounded"
/>
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ParameterValueType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Label" type="StringLocIDType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DefaultValueType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataSetReference" type="DataSetReferenceType" minOccurs="0" />
    <xsd:element name="Values" type="ValuesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ValuesType">
  <xsd:sequence>
    <xsd:element name="Value" type="xsd:string" minOccurs="1" maxOccurs="unbounded"
nillable="true" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataSetsType">
  <xsd:sequence>

```



```

        <xsd:element name="DataSet" type="DataSetType" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataSetType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Fields" type="FieldsType" minOccurs="0" />
        <xsd:element name="Query" type="QueryType" minOccurs="0" />
        <xsd:element name="SharedDataSet" type="SharedDataSetType" minOccurs="0" />
        <xsd:element name="CaseSensitivity" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="True" />
                    <xsd:enumeration value="False" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Collation" type="xsd:string" minOccurs="0" />
        <xsd:element name="AccentSensitivity" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="True" />
                    <xsd:enumeration value="False" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="KanatypeSensitivity" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="True" />
                    <xsd:enumeration value="False" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="WidthSensitivity" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="True" />
                    <xsd:enumeration value="False" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
        <xsd:element name="InterpretSubtotalsAsDetails" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="True" />
                    <xsd:enumeration value="False" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="FieldsType">
    <xsd:sequence>
        <xsd:element name="Field" type="FieldType" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="StringWithDataTypeAttribute">
    <xsd:simpleContent>

```

```

    <xsd:extension base="xsd:string">
      <xsd:attribute name="DataType" use="optional">
        <xsd:simpleType>
          <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Boolean" />
            <xsd:enumeration value="DateTime" />
            <xsd:enumeration value="Integer" />
            <xsd:enumeration value="Float" />
            <xsd:enumeration value="String" />
          </xsd:restriction>
        </xsd:simpleType>
      </xsd:attribute>
      <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="FieldType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataField" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="StringWithDataTypeAttribute" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="QueryType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataSourceName" type="xsd:string" />
    <xsd:element name="CommandType" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Text" />
          <xsd:enumeration value="StoredProcedure" />
          <xsd:enumeration value="TableDirect" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="CommandText" type="xsd:string" />
    <xsd:element name="QueryParameters" type="QueryParametersType" minOccurs="0" />
    <xsd:element name="Timeout" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="SharedDataSetType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="SharedDataSetReference" type="xsd:string" />
    <xsd:element name="QueryParameters" type="QueryParametersType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataSourcesType">
  <xsd:sequence>
    <xsd:element name="DataSource" type="DataSourceType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataSourceType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Transaction" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="ConnectionProperties" type="ConnectionPropertiesType"
minOccurs="0" />
    <xsd:element name="DataSourceReference" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

<xsd:complexType name="ConnectionPropertiesType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataProvider" type="xsd:string" />
    <xsd:element name="ConnectString" type="xsd:string" />
    <xsd:element name="IntegratedSecurity" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="Prompt" type="StringLocIDType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="QueryParametersType">
  <xsd:sequence>
    <xsd:element name="QueryParameter" type="QueryParameterType" maxOccurs="unbounded"
/>
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="QueryParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="StringWithDataAttribute" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="CodeModulesType">
  <xsd:sequence>
    <xsd:element name="CodeModule" type="xsd:string" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ClassesType">
  <xsd:sequence>
    <xsd:element name="Class" type="ClassType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ClassType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ClassName" type="xsd:string" />
    <xsd:element name="InstanceName" type="xsd:normalizedString" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="BodyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="1" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="PageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="PageHeader" type="PageSectionType" minOccurs="0" />
    <xsd:element name="PageFooter" type="PageSectionType" minOccurs="0" />
    <xsd:element name="PageHeight" type="SizeType" minOccurs="0" />
    <xsd:element name="PageWidth" type="SizeType" minOccurs="0" />
    <xsd:element name="InteractiveHeight" type="SizeType" minOccurs="0" />
    <xsd:element name="InteractiveWidth" type="SizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="RightMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="TopMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="BottomMargin" type="SizeType" minOccurs="0" />
    <xsd:element name="Columns" type="xsd:int" minOccurs="0" />
    <xsd:element name="ColumnSpacing" type="SizeType" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
  </xsd:choice>

```

```

        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="PageSectionType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Height" type="SizeType" />
        <xsd:element name="PrintOnFirstPage" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="PrintOnLastPage" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="PrintBetweenSections" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="EmbeddedImagesType">
    <xsd:sequence>
        <xsd:element name="EmbeddedImage" type="EmbeddedImageType" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="EmbeddedImageType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="MIMETYPE" type="xsd:string" />
        <xsd:element name="ImageData" type="xsd:string" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ReportItemsType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="Line" type="LineType" />
        <xsd:element name="Rectangle" type="RectangleType" />
        <xsd:element name="Textbox" type="TextboxType" />
        <xsd:element name="Image" type="ImageType" />
        <xsd:element name="Subreport" type="SubreportType" />
        <xsd:element name="Chart" type="ChartType" />
        <xsd:element name="GaugePanel" type="GaugePanelType" />
        <xsd:element name="Map" type="MapType" />
        <xsd:element name="Tablix" type="TablixType" />
        <xsd:element name="CustomReportItem" type="CustomReportItemType" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ActionInfoType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Actions" type="ActionsType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ActionsType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Action" type="ActionType" minOccurs="1" maxOccurs="unbounded" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ActionType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Hyperlink" type="xsd:string" minOccurs="0" />
        <xsd:element name="Drillthrough" type="DrillthroughType" minOccurs="0" />
        <xsd:element name="BookmarkLink" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />

```

```

</xsd:complexType>
<xsd:complexType name="DrillthroughType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ReportName" type="xsd:string" />
    <xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="VisibilityType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToggleItem" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="LineType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="RectangleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="LinkToChild" type="xsd:string" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
    <xsd:element name="PageName" type="xsd:string" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TextboxType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="Paragraphs" type="ParagraphsType" minOccurs="1" />
    <xsd:element name="CanGrow" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="CanShrink" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="HideDuplicates" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToggleImage" type="ToggleImageType" minOccurs="0" />
    <xsd:element name="UserSort" type="UserSortType" minOccurs="0" />
    <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementStyle" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Auto" />
          <xsd:enumeration value="Attribute" />
          <xsd:enumeration value="Element" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ParagraphsType">
  <xsd:sequence>
    <xsd:element name="Paragraph" type="ParagraphType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax" />

```

```

</xsd:complexType>
<xsd:complexType name="ParagraphType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TextRuns" type="TextRunsType" minOccurs="1" />
    <xsd:element name="LeftIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="HangingIndent" type="xsd:string" minOccurs="0" />
    <xsd:element name="ListStyle" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="None" />
          <xsd:enumeration value="Bulleted" />
          <xsd:enumeration value="Numbered" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="ListLevel" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="SpaceBefore" type="xsd:string" minOccurs="0" />
    <xsd:element name="SpaceAfter" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TextRunsType">
  <xsd:sequence>
    <xsd:element name="TextRun" type="TextRunType" minOccurs="1" maxOccurs="unbounded"
  />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TextRunType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Value" type="LocIDStringWithDataAttribute" minOccurs="1" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkupType" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ToggleImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="InitialState" type="xsd:string" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="Source">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="External" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

        <xsd:enumeration value="Embedded" />
        <xsd:enumeration value="Database" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="Value" type="xsd:string" />
<xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
<xsd:element name="Sizing" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="AutoSize" />
            <xsd:enumeration value="Fit" />
            <xsd:enumeration value="FitProportional" />
            <xsd:enumeration value="Clip" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Output" />
            <xsd:enumeration value="NoOutput" />
            <xsd:enumeration value="ContentsOnly" />
            <xsd:enumeration value="Auto" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="SubreportType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:element name="Top" type="SizeType" minOccurs="0" />
        <xsd:element name="Left" type="SizeType" minOccurs="0" />
        <xsd:element name="Height" type="SizeType" minOccurs="0" />
        <xsd:element name="Width" type="SizeType" minOccurs="0" />
        <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
        <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
        <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
        <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
        <xsd:element name="ReportName" type="xsd:string" />
        <xsd:element name="Parameters" type="ParametersType" minOccurs="0" />
        <xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
        <xsd:element name="MergeTransactions" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
        <xsd:element name="DataElementOutput" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Output" />
                    <xsd:enumeration value="NoOutput" />
                    <xsd:enumeration value="ContentsOnly" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />

```



```

</xsd:complexType>
<xsd:complexType name="CustomReportItemType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Type" type="xsd:string" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="AltReportItem" type="ReportItemsType" minOccurs="0" />
    <xsd:element name="CustomData" type="CustomDataType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="CustomDataType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
    <xsd:element name="DataColumnHierarchy" type="DataColumnHierarchyType" minOccurs="0" />
    <xsd:element name="DataRowHierarchy" type="DataRowHierarchyType" minOccurs="0" />
    <xsd:element name="DataRows" type="DataRowsType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataColumnHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataMembers" type="DataMembersType" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataRowHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DataMembers" type="DataMembersType" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataMembersType">
  <xsd:sequence>
    <xsd:element name="DataMember" type="DataMemberType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataMemberType">

```

```

<xsd:choice minOccurs="0" maxOccurs="unbounded">
  <xsd:element name="Group" type="GroupType" minOccurs="0" />
  <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
  <xsd:element name="Subtotal" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
  <xsd:element name="DataMembers" type="DataMembersType" minOccurs="0" />
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataRowsType">
  <xsd:sequence>
    <xsd:element name="DataRow" type="DataRowType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataRowType">
  <xsd:sequence>
    <xsd:element name="DataCell" type="DataCellType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataCellType">
  <xsd:sequence>
    <xsd:element name="DataValue" type="DataValueType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ParametersType">
  <xsd:sequence>
    <xsd:element name="Parameter" type="ParameterType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ParameterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="Omit" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:string" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GroupType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="GroupExpressions" type="GroupExpressionsType" minOccurs="0" />
    <xsd:element name="ReGroupExpressions" type="GroupExpressionsType" minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
    <xsd:element name="PageName" type="xsd:string" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="Parent" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Variables" type="VariablesType" minOccurs="0" />
    <xsd:element name="DomainScope" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

<xsd:complexType name="VariablesType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Variable" type="VariableType" minOccurs="1" maxOccurs="unbounded"
  />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="VariableType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="StringWithDataModelAttribute" minOccurs="1"
maxOccurs="1" />
    <xsd:element name="Writable" type="xsd:boolean" minOccurs="0" maxOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GroupExpressionsType">
  <xsd:sequence>
    <xsd:element name="GroupExpression" type="xsd:string" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="SortExpressionsType">
  <xsd:sequence>
    <xsd:element name="SortExpression" type="SortExpressionType" minOccurs="1"
maxOccurs="unbounded" />
    <xsd:any namespace="##other" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="SortExpressionType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="Direction" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Ascending" />
          <xsd:enumeration value="Descending" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
    <xsd:element name="PageName" type="xsd:string" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="ChartSeriesHierarchy" type="ChartHierarchyType" />
<xsd:element name="ChartCategoryHierarchy" type="ChartHierarchyType" />
<xsd:element name="ChartData" type="ChartDataTypes" minOccurs="0" />
<xsd:element name="ChartAreas" type="ChartAreasType" minOccurs="0" />
<xsd:element name="ChartLegends" type="ChartLegendsType" minOccurs="0" />
<xsd:element name="ChartTitles" type="ChartTitlesType" minOccurs="0" />
<xsd:element name="DynamicHeight" type="xsd:string" minOccurs="0" />
<xsd:element name="DynamicWidth" type="xsd:string" minOccurs="0" />
<xsd:element name="Palette" type="xsd:string" minOccurs="0" />
<xsd:element name="ChartCustomPaletteColors" type="ChartCustomPaletteColorsType"
minOccurs="0" />
<xsd:element name="PaletteHatchBehavior" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
      <xsd:enumeration value="Auto" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="ChartBorderSkin" type="ChartBorderSkinType" minOccurs="0" />
<xsd:element name="ChartNoDataMessage" type="ChartTitleType" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartHierarchyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ChartMembers" type="ChartMembersType" minOccurs="1" maxOccurs="1"
/>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartMembersType">
  <xsd:sequence minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="ChartMember" type="ChartMemberType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartMemberType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="0" maxOccurs="1" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0"
maxOccurs="1" />
    <xsd:element name="ChartMembers" type="ChartMembersType" minOccurs="0" maxOccurs="1"
/>
  </xsd:choice>
  <xsd:element name="Label" type="StringLocIDType" minOccurs="1" maxOccurs="1" />
  <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0"
maxOccurs="1" />
  <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
  <xsd:element name="DataElementOutput" minOccurs="0" maxOccurs="1">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Output" />
        <xsd:enumeration value="NoOutput" />
        <xsd:enumeration value="ContentsOnly" />
        <xsd:enumeration value="Auto" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />

```

```

</xsd:complexType>
<xsd:complexType name="ChartAreasType">
  <xsd:sequence>
    <xsd:element name="ChartArea" type="ChartAreaType" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartAreaType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartCategoryAxes" type="ChartCategoryAxesType" minOccurs="0" />
    <xsd:element name="ChartValueAxes" type="ChartValueAxesType" minOccurs="0" />
    <xsd:element name="ChartThreeDProperties" type="ChartThreeDPropertiesType"
minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="AlignOrientation" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartAlignType" type="ChartAlignTypeType" minOccurs="0" />
    <xsd:element name="ChartElementPosition" type="ChartElementPositionType"
minOccurs="0" />
    <xsd:element name="ChartInnerPlotPosition" type="ChartElementPositionType"
minOccurs="0" />
    <xsd:element name="AlignWithChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="EquallySizedAxesFont" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartAlignTypeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="AxesView" type="xsd:string" minOccurs="0" />
    <xsd:element name="Cursor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="InnerPlotPosition" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartElementPositionType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartTitlesType">
  <xsd:sequence>
    <xsd:element name="ChartTitle" type="ChartTitleType" minOccurs="0"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartTitleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Caption" type="StringLocIDType" minOccurs="1" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockToChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideChartArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="ChartElementPosition" type="ChartElementPositionType"
minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendsType">
    <xsd:sequence>
        <xsd:element name="ChartLegend" type="ChartLegendType" minOccurs="0"
maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="Position" type="xsd:string" minOccurs="0" />
        <xsd:element name="Layout" type="xsd:string" minOccurs="0" />
        <xsd:element name="DockToChartArea" type="xsd:string" minOccurs="0" />
        <xsd:element name="DockOutsideChartArea" type="xsd:string" minOccurs="0" />
        <xsd:element name="ChartElementPosition" type="ChartElementPositionType"
minOccurs="0" />
        <xsd:element name="ChartLegendTitle" type="ChartLegendTitleType" minOccurs="0" />
        <xsd:element name="AutoFitTextDisabled" type="xsd:string" minOccurs="0" />
        <xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
        <xsd:element name="ChartLegendColumns" type="ChartLegendColumnsType" minOccurs="0"
/>
        <xsd:element name="HeaderSeparator" type="xsd:string" minOccurs="0" />
        <xsd:element name="HeaderSeparatorColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="ColumnSeparator" type="xsd:string" minOccurs="0" />
        <xsd:element name="ColumnSeparatorColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="ColumnSpacing" type="xsd:string" minOccurs="0" />
        <xsd:element name="InterlacedRows" type="xsd:string" minOccurs="0" />
        <xsd:element name="InterlacedRowsColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="EquallySpacedItems" type="xsd:string" minOccurs="0" />
        <xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
        <xsd:element name="MaxAutoSize" type="xsd:string" minOccurs="0" />
        <xsd:element name="TextWrapThreshold" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendTitleType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Caption" type="StringLocIDType" />
        <xsd:element name="TitleSeparator" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartCustomPaletteColorsType">
    <xsd:sequence>
        <xsd:element name="ChartCustomPaletteColor" type="xsd:string" maxOccurs="unbounded"
/>
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartBorderSkinType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="ChartBorderSkinType" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
</xsd:complexType>
<xsd:complexType name="ChartLegendColumnsType">
    <xsd:sequence>

```

```

        <xsd:element name="ChartLegendColumn" type="ChartLegendColumnType"
maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendColumnType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="ColumnType">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Text" />
                    <xsd:enumeration value="SeriesSymbol" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="Value" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="MinimumWidth" type="xsd:string" minOccurs="0" />
        <xsd:element name="MaximumWidth" type="xsd:string" minOccurs="0" />
        <xsd:element name="SeriesSymbolWidth" type="xsd:string" minOccurs="0" />
        <xsd:element name="SeriesSymbolHeight" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendColumnHeaderType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Value" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendCustomItemsType">
    <xsd:sequence>
        <xsd:element name="ChartLegendCustomItem" type="ChartLegendCustomItemType"
maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendCustomItemType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="ChartLegendCustomItemCells" type="ChartLegendCustomItemCellsType"
/>
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
        <xsd:element name="Separator" type="xsd:string" minOccurs="0" />
        <xsd:element name="SeparatorColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendCustomItemCellsType">
    <xsd:sequence>
        <xsd:element name="ChartLegendCustomItemCell" type="ChartLegendCustomItemCellType"
maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartLegendCustomItemCellType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="CellType" minOccurs="0">
            <xsd:simpleType>

```

```

        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Text" />
            <xsd:enumeration value="SeriesSymbol" />
            <xsd:enumeration value="Image" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element name="Text" type="xsd:string" minOccurs="0" />
<xsd:element name="CellSpan" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
<xsd:element name="ImageHeight" type="xsd:string" minOccurs="0" />
<xsd:element name="ImageWidth" type="xsd:string" minOccurs="0" />
<xsd:element name="SymbolHeight" type="xsd:string" minOccurs="0" />
<xsd:element name="SymbolWidth" type="xsd:string" minOccurs="0" />
<xsd:element name="Alignment" type="xsd:string" minOccurs="0" />
<xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartMarkerType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Type" type="xsd:string" minOccurs="0" />
        <xsd:element name="Size" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartCategoryAxesType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="ChartAxis" type="ChartAxisType" maxOccurs="unbounded" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartValueAxesType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="ChartAxis" type="ChartAxisType" minOccurs="1" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartAxisType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Visible" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="ChartAxisTitle" type="ChartAxisTitleType" minOccurs="0" />
        <xsd:element name="Margin" type="xsd:string" minOccurs="0" />
        <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
        <xsd:element name="VariableAutoInterval" type="xsd:string" minOccurs="0" />
        <xsd:element name="LabelInterval" type="xsd:string" minOccurs="0" />
        <xsd:element name="LabelIntervalType" type="xsd:string" minOccurs="0" />
        <xsd:element name="LabelIntervalOffset" type="xsd:string" minOccurs="0" />
        <xsd:element name="LabelIntervalOffsetType" type="xsd:string" minOccurs="0" />
        <xsd:element name="ChartMajorGridLines" type="ChartGridLinesType" minOccurs="0" />
        <xsd:element name="ChartMinorGridLines" type="ChartGridLinesType" minOccurs="0" />
        <xsd:element name="ChartMajorTickMarks" type="ChartTickMarksType" minOccurs="0" />
        <xsd:element name="ChartMinorTickMarks" type="ChartTickMarksType" minOccurs="0" />
        <xsd:element name="MarksAlwaysAtPlotEdge" type="xsd:string" minOccurs="0" />
    </xsd:choice>

```



```

<xsd:element name="Reverse" type="xsd:string" minOccurs="0" />
<xsd:element name="CrossAt" type="xsd:string" minOccurs="0" />
<xsd:element name="Location" type="xsd:string" minOccurs="0" />
<xsd:element name="Interlaced" type="xsd:string" minOccurs="0" />
<xsd:element name="InterlacedColor" type="xsd:string" minOccurs="0" />
<xsd:element name="ChartStripLines" type="ChartStripLinesType" minOccurs="0" />
<xsd:element name="Arrows" type="xsd:string" minOccurs="0" />
<xsd:element name="Scalar" type="xsd:boolean" minOccurs="0" />
<xsd:element name="Minimum" type="xsd:string" minOccurs="0" />
<xsd:element name="Maximum" type="xsd:string" minOccurs="0" />
<xsd:element name="LogScale" type="xsd:string" minOccurs="0" />
<xsd:element name="LogBase" type="xsd:string" minOccurs="0" />
<xsd:element name="HideLabels" type="xsd:string" minOccurs="0" />
<xsd:element name="Angle" type="xsd:string" minOccurs="0" />
<xsd:element name="PreventFontShrink" type="xsd:string" minOccurs="0" />
<xsd:element name="PreventFontGrow" type="xsd:string" minOccurs="0" />
<xsd:element name="PreventLabelOffset" type="xsd:string" minOccurs="0" />
<xsd:element name="PreventWordWrap" type="xsd:string" minOccurs="0" />
<xsd:element name="AllowLabelRotation" type="xsd:string" minOccurs="0" />
<xsd:element name="IncludeZero" type="xsd:string" minOccurs="0" />
<xsd:element name="LabelsAutoFitDisabled" type="xsd:string" minOccurs="0" />
<xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
<xsd:element name="MaxFontSize" type="xsd:string" minOccurs="0" />
<xsd:element name="OffsetLabels" type="xsd:string" minOccurs="0" />
<xsd:element name="HideEndLabels" type="xsd:string" minOccurs="0" />
<xsd:element name="ChartAxisScaleBreak" type="ChartAxisScaleBreakType" minOccurs="0" />
/>
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartAxisTitleType">
<xsd:choice maxOccurs="unbounded">
<xsd:element name="Caption" type="StringLocIDType" />
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartStripLinesType">
<xsd:sequence>
<xsd:element name="ChartStripLine" type="ChartStripLineType" maxOccurs="unbounded" />
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartStripLineType">
<xsd:choice minOccurs="0" maxOccurs="unbounded">
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="Title" type="xsd:string" minOccurs="0" />
<xsd:element name="TextOrientation" type="xsd:string" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
<xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
<xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
<xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
<xsd:element name="StripWidth" type="xsd:string" minOccurs="0" />
<xsd:element name="StripWidthType" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartAxisScaleBreakType">
<xsd:choice minOccurs="0" maxOccurs="unbounded">

```

```

        <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
        <xsd:element name="BreakLineType" type="xsd:string" minOccurs="0" />
        <xsd:element name="CollapsibleSpaceThreshold" type="xsd:string" minOccurs="0" />
        <xsd:element name="MaxNumberOfBreaks" type="xsd:string" minOccurs="0" />
        <xsd:element name="Spacing" type="xsd:string" minOccurs="0" />
        <xsd:element name="IncludeZero" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartDataTypes">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="ChartSeriesCollection" type="ChartSeriesCollectionType"
minOccurs="1" maxOccurs="1" />
        <xsd:element name="ChartDerivedSeriesCollection"
type="ChartDerivedSeriesCollectionType" minOccurs="0" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartSeriesCollectionType">
    <xsd:sequence maxOccurs="unbounded" minOccurs="1">
        <xsd:element name="ChartSeries" type="ChartSeriesType" minOccurs="1"
maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartDerivedSeriesCollectionType">
    <xsd:sequence maxOccurs="unbounded" minOccurs="1">
        <xsd:element name="ChartDerivedSeries" type="ChartDerivedSeriesType" minOccurs="1"
maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartSeriesType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
        <xsd:element name="ChartDataPoints" type="ChartDataPointsType" minOccurs="0" />
        <xsd:element name="Type" type="xsd:string" minOccurs="0" />
        <xsd:element name="Subtype" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="ChartEmptyPoints" type="ChartEmptyPointsType" minOccurs="0" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
        <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
        <xsd:element name="ChartItemInLegend" type="ChartItemInLegendType" minOccurs="0" />
        <xsd:element name="ChartAreaName" type="xsd:string" minOccurs="0" />
        <xsd:element name="ValueAxisName" type="xsd:string" minOccurs="0" />
        <xsd:element name="CategoryAxisName" type="xsd:string" minOccurs="0" />
        <xsd:element name="ChartSmartLabel" type="ChartSmartLabelType" minOccurs="0" />
        <xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
        <xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartDerivedSeriesType">
    <xsd:choice minOccurs="3" maxOccurs="unbounded">
        <xsd:element name="ChartSeries" type="ChartSeriesType" minOccurs="1" />
        <xsd:element name="SourceChartSeriesName" type="xsd:string" minOccurs="1" />
        <xsd:element name="DerivedSeriesFormula" minOccurs="1">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="BollingerBands" />
                    <xsd:enumeration value="MovingAverage" />
                    <xsd:enumeration value="ExponentialMovingAverage" />
                    <xsd:enumeration value="TriangularMovingAverage" />
                    <xsd:enumeration value="WeightedMovingAverage" />
                    <xsd:enumeration value="MACD" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
    </xsd:choice>

```

```

        <xsd:enumeration value="DetrendedPriceOscillator" />
        <xsd:enumeration value="Envelopes" />
        <xsd:enumeration value="Performance" />
        <xsd:enumeration value="RateOfChange" />
        <xsd:enumeration value="RelativeStrengthIndex" />
        <xsd:enumeration value="StandardDeviation" />
        <xsd:enumeration value="TRIX" />
        <xsd:enumeration value="Mean" />
        <xsd:enumeration value="Median" />
    </xsd:restriction>
</xsd:simpleType>
</xsd:element>
<xsd:element name="ChartFormulaParameters" type="ChartFormulaParametersType"
minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartFormulaParametersType">
<xsd:sequence>
<xsd:element name="ChartFormulaParameter" type="ChartFormulaParameterType"
maxOccurs="unbounded" />
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartFormulaParameterType">
<xsd:choice minOccurs="1">
<xsd:element name="Value" type="xsd:string" minOccurs="0" />
<xsd:element name="Source" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:string" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartEmptyPointsType">
<xsd:choice minOccurs="0" maxOccurs="unbounded">
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
<xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
<xsd:element name="AxisLabel" type="xsd:string" minOccurs="0" />
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartItemInLegendType">
<xsd:choice minOccurs="0" maxOccurs="unbounded">
<xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartDataPointsType">
<xsd:sequence>
<xsd:element name="ChartDataPoint" type="ChartDataPointType" maxOccurs="unbounded"
/>
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartDataPointType">
<xsd:choice minOccurs="0" maxOccurs="unbounded">
<xsd:element name="ChartDataPointValues" type="ChartDataPointValuesType"
minOccurs="0" />
<xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />

```

```

<xsd:element name="AxisLabel" type="xsd:string" minOccurs="0" />
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
<xsd:element name="ChartItemInLegend" type="ChartItemInLegendType" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
      <xsd:enumeration value="ContentsOnly" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartDataPointValuesType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="X" type="xsd:string" minOccurs="0" />
    <xsd:element name="Y" type="xsd:string" minOccurs="0" />
    <xsd:element name="Size" type="xsd:string" minOccurs="0" />
    <xsd:element name="High" type="xsd:string" minOccurs="0" />
    <xsd:element name="Low" type="xsd:string" minOccurs="0" />
    <xsd:element name="Start" type="xsd:string" minOccurs="0" />
    <xsd:element name="End" type="xsd:string" minOccurs="0" />
    <xsd:element name="Mean" type="xsd:string" minOccurs="0" />
    <xsd:element name="Median" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="DataValueType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartDataLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Visible" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Label" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="UseValueAsLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="Rotation" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartSmartLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Disabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowOutsidePlotArea" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutBackColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineAnchor" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutLineWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="CalloutStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowOverlapped" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

        <xsd:element name="MarkerOverlapping" type="xsd:string" minOccurs="0" />
        <xsd:element name="MaxMovingDistance" type="xsd:string" minOccurs="0" />
        <xsd:element name="MinMovingDistance" type="xsd:string" minOccurs="0" />
        <xsd:element name="ChartNoMoveDirections" type="ChartNoMoveDirectionsType"
minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartNoMoveDirectionsType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Up" type="xsd:string" minOccurs="0" />
        <xsd:element name="Left" type="xsd:string" minOccurs="0" />
        <xsd:element name="Right" type="xsd:string" minOccurs="0" />
        <xsd:element name="Down" type="xsd:string" minOccurs="0" />
        <xsd:element name="UpLeft" type="xsd:string" minOccurs="0" />
        <xsd:element name="UpRight" type="xsd:string" minOccurs="0" />
        <xsd:element name="DownLeft" type="xsd:string" minOccurs="0" />
        <xsd:element name="DownRight" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartThreeDPropertiesType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
        <xsd:element name="ProjectionMode" type="xsd:string" minOccurs="0" />
        <xsd:element name="Rotation" type="xsd:string" minOccurs="0" />
        <xsd:element name="Inclination" type="xsd:string" minOccurs="0" />
        <xsd:element name="Perspective" type="xsd:string" minOccurs="0" />
        <xsd:element name="DepthRatio" type="xsd:string" minOccurs="0" />
        <xsd:element name="Shading" type="xsd:string" minOccurs="0" />
        <xsd:element name="GapDepth" type="xsd:string" minOccurs="0" />
        <xsd:element name="WallThickness" type="xsd:string" minOccurs="0" />
        <xsd:element name="Clustered" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartGridLinesType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ChartTickMarksType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Enabled" type="xsd:string" minOccurs="0" />
        <xsd:element name="Type" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="Length" type="xsd:string" minOccurs="0" />
        <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalType" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
        <xsd:element name="IntervalOffsetType" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="StyleType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <xsd:element name="Border" type="BorderType" minOccurs="0" />
        <xsd:element name="TopBorder" type="BorderType" minOccurs="0" />

```

```

<xsd:element name="BottomBorder" type="BorderType" minOccurs="0" />
<xsd:element name="LeftBorder" type="BorderType" minOccurs="0" />
<xsd:element name="RightBorder" type="BorderType" minOccurs="0" />
<xsd:element name="BackgroundColor" type="xsd:string" minOccurs="0" />
<xsd:element name="BackgroundGradientType" type="xsd:string" minOccurs="0" />
<xsd:element name="BackgroundGradientEndColor" type="xsd:string" minOccurs="0" />
<xsd:element name="BackgroundHatchType" type="xsd:string" minOccurs="0" />
<xsd:element name="BackgroundImage" type="BackgroundImageType" minOccurs="0" />
<xsd:element name="FontStyle" type="xsd:string" minOccurs="0" />
<xsd:element name="FontFamily" type="xsd:string" minOccurs="0" />
<xsd:element name="FontSize" type="xsd:string" minOccurs="0" />
<xsd:element name="FontWeight" type="xsd:string" minOccurs="0" />
<xsd:element name="Format" type="xsd:string" minOccurs="0" />
<xsd:element name="TextDecoration" type="xsd:string" minOccurs="0" />
<xsd:element name="TextAlign" type="xsd:string" minOccurs="0" />
<xsd:element name="TextEffect" type="xsd:string" minOccurs="0" />
<xsd:element name="VerticalAlign" type="xsd:string" minOccurs="0" />
<xsd:element name="Color" type="xsd:string" minOccurs="0" />
<xsd:element name="ShadowColor" type="xsd:string" minOccurs="0" />
<xsd:element name="ShadowOffset" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingLeft" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingRight" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingTop" type="xsd:string" minOccurs="0" />
<xsd:element name="PaddingBottom" type="xsd:string" minOccurs="0" />
<xsd:element name="LineHeight" type="xsd:string" minOccurs="0" />
<xsd:element name="Direction" type="xsd:string" minOccurs="0" />
<xsd:element name="WritingMode" type="xsd:string" minOccurs="0" />
<xsd:element name="Language" type="xsd:string" minOccurs="0" />
<xsd:element name="UnicodeBiDi" type="xsd:string" minOccurs="0" />
<xsd:element name="Calendar" type="xsd:string" minOccurs="0" />
<xsd:element name="NumeralLanguage" type="xsd:string" minOccurs="0" />
<xsd:element name="NumeralVariant" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="BorderType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Color" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="BackgroundImageType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Source">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="External" />
          <xsd:enumeration value="Embedded" />
          <xsd:enumeration value="Database" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="Value" type="xsd:string" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="BackgroundRepeat" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="FiltersType">
  <xsd:sequence>
    <xsd:element name="Filter" type="FilterType" maxOccurs="unbounded" />
  </xsd:sequence>

```

```

    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="FilterType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="FilterExpression" type="xsd:string" />
    <xsd:element name="Operator">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Equal" />
          <xsd:enumeration value="Like" />
          <xsd:enumeration value="NotEqual" />
          <xsd:enumeration value="GreaterThan" />
          <xsd:enumeration value="GreaterThanOrEqual" />
          <xsd:enumeration value="LessThan" />
          <xsd:enumeration value="LessThanOrEqual" />
          <xsd:enumeration value="TopN" />
          <xsd:enumeration value="BottomN" />
          <xsd:enumeration value="TopPercent" />
          <xsd:enumeration value="BottomPercent" />
          <xsd:enumeration value="In" />
          <xsd:enumeration value="Between" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="FilterValues" type="FilterValuesType" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="FilterValuesType">
  <xsd:sequence>
    <xsd:element name="FilterValue" type="StringWithDataTypeAttribute"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="UserSortType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="SortExpression" type="xsd:string" />
    <xsd:element name="SortExpressionScope" type="xsd:string" minOccurs="0" />
    <xsd:element name="SortTarget" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:simpleType name="SizeType">
  <xsd:restriction base="xsd:normalizedString">
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType name="StringLocIDType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:complexType name="LocIDStringWithDataTypeAttribute">
  <xsd:simpleContent>
    <xsd:extension base="StringWithDataTypeAttribute">
      <xsd:attribute name="EvaluationMode" type="EvaluationModeType" default="Auto" />
      <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
<xsd:simpleType name="EvaluationModeType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Auto" />
    <xsd:enumeration value="Constant" />
  </xsd:restriction>

```

```

</xsd:simpleType>
<xsd:complexType name="CustomPropertiesType">
  <xsd:sequence>
    <xsd:element name="CustomProperty" type="CustomPropertyType" maxOccurs="unbounded"
  />
</xsd:sequence>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="CustomPropertyType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" />
    <xsd:element name="Value" type="xsd:string" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixType">
  <xsd:choice minOccurs="1" maxOccurs="unbounded">
    <xsd:element name="TablixCorner" type="TablixCornerType" minOccurs="0" />
    <xsd:element name="TablixBody" type="TablixBodyType" minOccurs="0" />
    <xsd:element name="TablixColumnHierarchy" type="TablixHierarchyType" minOccurs="1"
  />
  <xsd:element name="TablixRowHierarchy" type="TablixHierarchyType" minOccurs="1" />
  <xsd:element name="LayoutDirection" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="LTR" />
        <xsd:enumeration value="RTL" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:element name="GroupsBeforeRowHeaders" type="xsd:unsignedInt" minOccurs="0" />
  <xsd:element name="RepeatColumnHeaders" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="RepeatRowHeaders" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="FixedColumnHeaders" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="FixedRowHeaders" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="Style" type="StyleType" minOccurs="0" />
  <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
  <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
  <xsd:element name="Top" type="SizeType" minOccurs="0" />
  <xsd:element name="Left" type="SizeType" minOccurs="0" />
  <xsd:element name="Height" type="SizeType" minOccurs="0" />
  <xsd:element name="Width" type="SizeType" minOccurs="0" />
  <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
  <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
  <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
  <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
  <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
  <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
  <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
  <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
  <xsd:element name="PageName" type="xsd:string" minOccurs="0" />
  <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
  <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
  <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
  <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
  <xsd:element name="OmitBorderOnPageBreak" type="xsd:boolean" minOccurs="0" />
  <xsd:element name="DataElementOutput" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Output" />
        <xsd:enumeration value="NoOutput" />
        <xsd:enumeration value="ContentsOnly" />
        <xsd:enumeration value="Auto" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <xsd:any namespace="##other" processContents="lax" />

```



```

    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
  <xsd:complexType name="TablixBodyType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="TablixColumns" type="TablixColumnsType" minOccurs="1"
maxOccurs="1" />
      <xsd:element name="TablixRows" type="TablixRowsType" minOccurs="1" maxOccurs="1" />
      <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
  <xsd:complexType name="TablixCornerType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="TablixCornerRows" type="TablixCornerRowsType" minOccurs="1"
maxOccurs="1" />
      <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
  <xsd:complexType name="TablixCornerRowsType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="TablixCornerRow" type="TablixCornerRowType" minOccurs="1"
maxOccurs="unbounded">
        </xsd:element>
        <xsd:any namespace="##other" processContents="lax" />
      </xsd:choice>
      <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
  <xsd:complexType name="TablixCornerRowType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="TablixCornerCell" type="TablixCornerCellType" minOccurs="0"
maxOccurs="unbounded">
        </xsd:element>
        <xsd:any namespace="##other" processContents="lax" />
      </xsd:choice>
      <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
  <xsd:complexType name="TablixCornerCellType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="CellContents" type="CellContentsType" minOccurs="0" maxOccurs="1"
/>
      <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
  <xsd:complexType name="TablixHierarchyType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1"
maxOccurs="1" />
      <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
  <xsd:complexType name="TablixMembersType">
    <xsd:sequence minOccurs="1" maxOccurs="unbounded">
      <xsd:element name="TablixMember" type="TablixMemberType" minOccurs="1"
maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
  <xsd:complexType name="TablixMemberType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
      <xsd:element name="Group" type="GroupType" minOccurs="0" maxOccurs="1" />
      <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0"
maxOccurs="1" />
      <xsd:element name="TablixHeader" type="TablixHeaderType" minOccurs="0" maxOccurs="1"
/>
    </xsd:choice>
  </xsd:complexType>

```

```

        <xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="FixedData" type="xsd:boolean" minOccurs="0" maxOccurs="1" />
        <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" maxOccurs="1" />
        <xsd:element name="HideIfNoRows" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="RepeatOnNewPage" type="xsd:boolean" minOccurs="0" />
        <xsd:element name="KeepWithGroup" minOccurs="0" maxOccurs="1">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="None" />
                    <xsd:enumeration value="Before" />
                    <xsd:enumeration value="After" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
        <xsd:element name="DataElementOutput" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Output" />
                    <xsd:enumeration value="NoOutput" />
                    <xsd:enumeration value="ContentsOnly" />
                    <xsd:enumeration value="Auto" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="KeepTogether" type="xsd:boolean" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixHeaderType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="Size" type="SizeType" minOccurs="1" maxOccurs="1" />
        <xsd:element name="CellContents" type="CellContentsType" minOccurs="1" maxOccurs="1"
/>
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="CellContentsType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="ColSpan" type="xsd:unsignedInt" minOccurs="0" maxOccurs="1">
            </xsd:element>
        <xsd:element name="RowSpan" type="xsd:unsignedInt" minOccurs="0" maxOccurs="1">
            </xsd:element>
        <xsd:element name="Line" type="LineType" minOccurs="0" maxOccurs="1" />
        <xsd:element name="Rectangle" type="RectangleType" minOccurs="0" maxOccurs="1" />
        <xsd:element name="Textbox" type="TextboxType" minOccurs="0" maxOccurs="1" />
        <xsd:element name="Image" type="ImageType" minOccurs="0" maxOccurs="1" />
        <xsd:element name="Subreport" type="SubreportType" minOccurs="0" maxOccurs="1" />
        <xsd:element name="Chart" type="ChartType" minOccurs="0" maxOccurs="1" />
        <xsd:element name="GaugePanel" type="GaugePanelType" minOccurs="0" maxOccurs="1" />
        <xsd:element name="Map" type="MapType" minOccurs="0" maxOccurs="1" />
        <xsd:element name="CustomReportItem" type="CustomReportItemType" minOccurs="0"
maxOccurs="1" />
        <xsd:element name="Tablix" type="TablixType" minOccurs="0" maxOccurs="1" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TablixColumnsType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="TablixColumn" type="TablixColumnType" minOccurs="1"
maxOccurs="unbounded" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>

```

```

        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
    <xsd:complexType name="TablixColumnType">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="Width" type="SizeType" minOccurs="1" maxOccurs="1" />
            <xsd:any namespace="##other" processContents="lax" />
        </xsd:choice>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
    <xsd:complexType name="TablixRowsType">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="TablixRow" type="TablixRowType" minOccurs="1"
maxOccurs="unbounded" />
            <xsd:any namespace="##other" processContents="lax" />
        </xsd:choice>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
    <xsd:complexType name="TablixRowType">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="Height" type="SizeType" minOccurs="1" maxOccurs="1" />
            <xsd:element name="TablixCells" type="TablixCellsType" minOccurs="1" maxOccurs="1"
/>
            <xsd:any namespace="##other" processContents="lax" />
        </xsd:choice>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
    <xsd:complexType name="TablixCellsType">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="TablixCell" type="TablixCellType" minOccurs="1"
maxOccurs="unbounded" />
            <xsd:any namespace="##other" processContents="lax" />
        </xsd:choice>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
    <xsd:complexType name="TablixCellType">
        <xsd:choice maxOccurs="unbounded">
            <xsd:element name="CellContents" type="CellContentsType" minOccurs="0" maxOccurs="1"
/>
            <xsd:any namespace="##other" processContents="lax" />
        </xsd:choice>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" maxOccurs="1" />
    <xsd:element name="DataElementOutput" minOccurs="0" maxOccurs="1">
        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="Output" />
                <xsd:enumeration value="NoOutput" />
                <xsd:enumeration value="ContentsOnly" />
                <xsd:enumeration value="Auto" />
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="PageBreakType">
    <xsd:choice minOccurs="1" maxOccurs="unbounded">
        <xsd:element name="Disabled" type="xsd:string" minOccurs="0" />
        <xsd:element name="ResetPageNumber" type="xsd:string" minOccurs="0" />
        <xsd:element name="BreakLocation" minOccurs="1">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="None" />
                    <xsd:enumeration value="Start" />
                    <xsd:enumeration value="End" />
                    <xsd:enumeration value="StartAndEnd" />
                    <xsd:enumeration value="Between" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>

```

```

</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GaugePanelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--DataRegionTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="Top" type="SizeType" minOccurs="0" />
    <xsd:element name="Left" type="SizeType" minOccurs="0" />
    <xsd:element name="Height" type="SizeType" minOccurs="0" />
    <xsd:element name="Width" type="SizeType" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
    <xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
    <xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
    <xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
    <xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
    <xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
    <xsd:element name="NoRowsMessage" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
    <xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
    <xsd:element name="PageName" type="xsd:string" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
          <xsd:enumeration value="ContentsOnly" />
          <xsd:enumeration value="Auto" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--DataRegionTypeEnd-->
    <xsd:element name="AntiAliasing" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextAntiAliasingQuality" type="xsd:string" minOccurs="0" />
    <xsd:element name="AutoLayout" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShadowIntensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="RadialGauges" type="RadialGaugesType" minOccurs="0" />
    <xsd:element name="LinearGauges" type="LinearGaugesType" minOccurs="0" />
    <xsd:element name="NumericIndicators" type="NumericIndicatorsType" minOccurs="0" />
    <xsd:element name="StateIndicators" type="StateIndicatorsType" minOccurs="0" />
    <xsd:element name="GaugeImages" type="GaugeImagesType" minOccurs="0" />
    <xsd:element name="GaugeLabels" type="GaugeLabelsType" minOccurs="0" />
    <xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
    <xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="GaugeMember" type="GaugeMemberType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GaugeMemberType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="1" />
    <xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
    <xsd:element name="GaugeMember" type="GaugeMemberType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GaugeInputValueType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="Formula" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinPercent" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="MaxPercent" type="xsd:string" minOccurs="0" />
<xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
<xsd:element name="AddConstant" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="RadialGaugeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeTypeStart-->
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
    <xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="ClipContent" type="xsd:string" minOccurs="0" />
    <xsd:element name="AspectRatio" type="xsd:string" minOccurs="0" />
    <!--GaugeTypeEnd-->
    <xsd:element name="GaugeScales" type="RadialScalesType" minOccurs="0" />
    <xsd:element name="PivotX" type="xsd:string" minOccurs="0" />
    <xsd:element name="PivotY" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="LinearGaugeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeTypeStart-->
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="BackFrame" type="BackFrameType" minOccurs="0" />
    <xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="ClipContent" type="xsd:string" minOccurs="0" />
    <xsd:element name="AspectRatio" type="xsd:string" minOccurs="0" />
    <!--GaugeTypeEnd-->
    <xsd:element name="GaugeScales" type="LinearScalesType" minOccurs="0" />
    <xsd:element name="Orientation" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

<xsd:complexType name="NumericIndicatorType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="NumericIndicatorRanges" type="NumericIndicatorRangesType"
minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="DecimalDigitColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="DecimalDigits" type="xsd:string" minOccurs="0" />
    <xsd:element name="DigitColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Digits" type="xsd:string" minOccurs="0" />
    <xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="LedDimColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffString" type="xsd:string" minOccurs="0" />
    <xsd:element name="OutOfRangeString" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="SeparatorWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowDecimalPoint" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowLeadingZeros" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowSign" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="StateIndicatorType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePanelItemTypeStart-->
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Height" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
    <!--GaugePanelItemTypeEnd-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="IndicatorImage" type="IndicatorImageType" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="TransformationType" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransformationScope" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="IndicatorStates" type="IndicatorStatesType" minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
    <xsd:element name="ScaleFactor" type="xsd:string" minOccurs="0" />
    <xsd:element name="StateDataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="StateDataElementOutput" minOccurs="0">

```

```

        <xsd:simpleType>
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="Output" />
                <xsd:enumeration value="NoOutput" />
            </xsd:restriction>
        </xsd:simpleType>
    </xsd:element>
    <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="GaugeImageType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <!--GaugePanelItemTypeStart-->
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
        <xsd:element name="Top" type="xsd:string" minOccurs="0" />
        <xsd:element name="Left" type="xsd:string" minOccurs="0" />
        <xsd:element name="Height" type="xsd:string" minOccurs="0" />
        <xsd:element name="Width" type="xsd:string" minOccurs="0" />
        <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
        <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
        <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
        <!--GaugePanelItemTypeEnd-->
        <xsd:element name="Source" type="xsd:string" minOccurs="1" />
        <xsd:element name="Value" type="xsd:string" minOccurs="1" />
        <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
        <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
        <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
        <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
        <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="GaugeLabelType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <!--GaugePanelItemTypeStart-->
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
        <xsd:element name="Top" type="xsd:string" minOccurs="0" />
        <xsd:element name="Left" type="xsd:string" minOccurs="0" />
        <xsd:element name="Height" type="xsd:string" minOccurs="0" />
        <xsd:element name="Width" type="xsd:string" minOccurs="0" />
        <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
        <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
        <xsd:element name="ParentItem" type="xsd:string" minOccurs="0" />
        <!--GaugePanelItemTypeEnd-->
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="Text" type="xsd:string" minOccurs="0" />
        <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
        <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
        <xsd:element name="TextShadowOffset" type="xsd:string" minOccurs="0" />
        <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="RadialScaleType">
    <xsd:choice minOccurs="0" maxOccurs="unbounded">
        <!--GaugeScaleTypeStart-->
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    </xsd:choice>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

<xsd:element name="ScaleRanges" type="ScaleRangesType" minOccurs="0" />
<xsd:element name="ScaleLabels" type="ScaleLabelsType" minOccurs="0" />
<xsd:element name="GaugeMajorTickMarks" type="GaugeTickMarksType" minOccurs="0" />
<xsd:element name="GaugeMinorTickMarks" type="GaugeTickMarksType" minOccurs="0" />
<xsd:element name="CustomLabels" type="CustomLabelsType" minOccurs="0" />
<xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
<xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
<xsd:element name="MaximumPin" type="ScalePinType" minOccurs="0" />
<xsd:element name="MinimumPin" type="ScalePinType" minOccurs="0" />
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
<xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
<xsd:element name="Logarithmic" type="xsd:string" minOccurs="0" />
<xsd:element name="LogarithmicBase" type="xsd:string" minOccurs="0" />
<xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
<xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
<xsd:element name="TickMarksOnTop" type="xsd:string" minOccurs="0" />
<!--GaugeScaleTypeEnd-->
<xsd:element name="GaugePointers" type="RadialPointersType" minOccurs="0" />
<xsd:element name="Radius" type="xsd:string" minOccurs="0" />
<xsd:element name="StartAngle" type="xsd:string" minOccurs="0" />
<xsd:element name="SweepAngle" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="LinearScaleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugeScaleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="ScaleRanges" type="ScaleRangesType" minOccurs="0" />
    <xsd:element name="ScaleLabels" type="ScaleLabelsType" minOccurs="0" />
    <xsd:element name="GaugeMajorTickMarks" type="GaugeTickMarksType" minOccurs="0" />
    <xsd:element name="GaugeMinorTickMarks" type="GaugeTickMarksType" minOccurs="0" />
    <xsd:element name="CustomLabels" type="CustomLabelsType" minOccurs="0" />
    <xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="MaximumPin" type="ScalePinType" minOccurs="0" />
    <xsd:element name="MinimumPin" type="ScalePinType" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="Logarithmic" type="xsd:string" minOccurs="0" />
    <xsd:element name="LogarithmicBase" type="xsd:string" minOccurs="0" />
    <xsd:element name="Multiplier" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reversed" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="TickMarksOnTop" type="xsd:string" minOccurs="0" />
    <!--GaugeScaleTypeEnd-->
    <xsd:element name="GaugePointers" type="LinearPointersType" minOccurs="0" />
    <xsd:element name="StartMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="RadialPointerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePointerTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="0" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```



```

<xsd:element name="PointerImage" type="PointerImageType" minOccurs="0" />
<xsd:element name="BarStart" type="xsd:string" minOccurs="0" />
<xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
<xsd:element name="MarkerLength" type="xsd:string" minOccurs="0" />
<xsd:element name="MarkerStyle" type="xsd:string" minOccurs="0" />
<xsd:element name="Placement" type="xsd:string" minOccurs="0" />
<xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
<xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
<xsd:element name="Type" type="xsd:string" minOccurs="0" />
<!--GaugePointerTypeEnd-->
<xsd:element name="PointerCap" type="PointerCapType" minOccurs="0" />
<xsd:element name="NeedleStyle" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="LinearPointerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--GaugePointerTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="PointerImage" type="PointerImageType" minOccurs="0" />
    <xsd:element name="BarStart" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerLength" type="xsd:string" minOccurs="0" />
    <xsd:element name="MarkerStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingEnabled" type="xsd:string" minOccurs="0" />
    <xsd:element name="SnappingInterval" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Type" type="xsd:string" minOccurs="0" />
    <!--GaugePointerTypeEnd-->
    <xsd:element name="Thermometer" type="ThermometerType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ThermometerType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="BulbOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="BulbSize" type="xsd:string" minOccurs="0" />
    <xsd:element name="ThermometerStyle" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="PointerCapType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="CapImage" type="CapImageType" minOccurs="0" />
    <xsd:element name="OnTop" type="xsd:string" minOccurs="0" />
    <xsd:element name="Reflection" type="xsd:string" minOccurs="0" />
    <xsd:element name="CapStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="NumericIndicatorRangeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">

```

```

    <xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="DecimalDigitColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="DigitColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="IndicatorStateType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="Color" type="xsd:string" minOccurs="1" />
    <xsd:element name="ScaleFactor" type="xsd:string" minOccurs="1" />
    <xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="1" />
    <xsd:element name="IndicatorImage" type="IndicatorImageType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ScaleRangeType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="BackgroundGradientType" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
    <xsd:element name="StartWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="InRangeBarPointerColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="InRangeLabelColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="InRangeTickMarksColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ScaleLabelsType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="RotateLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowEndLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="CustomLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkStyle" type="TickMarkStyleType" minOccurs="0" />
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

    <xsd:element name="RotateLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TickMarkStyleType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
    <xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shape" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GaugeTickMarksType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--TickMarkStyleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
    <xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shape" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <!--TickMarkStyleTypeEnd-->
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="IntervalOffset" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="ScalePinType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--TickMarkStyleTypeStart-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="EnableGradient" type="xsd:string" minOccurs="0" />
    <xsd:element name="GradientDensity" type="xsd:string" minOccurs="0" />
    <xsd:element name="Length" type="xsd:string" minOccurs="0" />
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="Shape" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <!--TickMarkStyleTypeEnd-->
    <xsd:element name="Location" type="xsd:string" minOccurs="0" />
    <xsd:element name="Enable" type="xsd:string" minOccurs="0" />
    <xsd:element name="PinLabel" type="PinLabelType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="PinLabelType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />

```

```

    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="AllowUpsideDown" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistanceFromScale" type="xsd:string" minOccurs="0" />
    <xsd:element name="FontAngle" type="xsd:string" minOccurs="0" />
    <xsd:element name="Placement" type="xsd:string" minOccurs="0" />
    <xsd:element name="RotateLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseFontPercent" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="TopImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="IndicatorImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="PointerImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="SizeType" minOccurs="0" />
    <xsd:element name="OffsetY" type="SizeType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="CapImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMETYPE" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="SizeType" minOccurs="0" />
    <xsd:element name="OffsetY" type="SizeType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

<xsd:complexType name="FrameImageType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <!--BaseGaugeImageTypeStart-->
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="Value" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <!--BaseGaugeImageTypeEnd-->
    <xsd:element name="HueColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <xsd:element name="ClipImage" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="BackFrameType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="FrameBackground" type="FrameBackgroundType" minOccurs="0" />
    <xsd:element name="FrameImage" type="FrameImageType" minOccurs="0" />
    <xsd:element name="FrameStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="FrameShape" type="xsd:string" minOccurs="0" />
    <xsd:element name="FrameWidth" type="xsd:string" minOccurs="0" />
    <xsd:element name="GlassEffect" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="FrameBackgroundType">
  <xsd:choice minOccurs="0" maxOccurs="unbounded">
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
  </xsd:choice>
</xsd:complexType>
<xsd:complexType name="RadialGaugesType">
  <xsd:sequence>
    <xsd:element name="RadialGauge" type="RadialGaugeType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="LinearGaugesType">
  <xsd:sequence>
    <xsd:element name="LinearGauge" type="LinearGaugeType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="NumericIndicatorsType">
  <xsd:sequence>
    <xsd:element name="NumericIndicator" type="NumericIndicatorType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="StateIndicatorsType">
  <xsd:sequence>
    <xsd:element name="StateIndicator" type="StateIndicatorType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GaugeImagesType">
  <xsd:sequence>
    <xsd:element name="GaugeImage" type="GaugeImageType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
<xsd:complexType name="GaugeLabelsType">

```

```

        <xsd:sequence>
            <xsd:element name="GaugeLabel" type="GaugeLabelType" minOccurs="1"
maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
    <xsd:complexType name="RadialScalesType">
        <xsd:sequence>
            <xsd:element name="RadialScale" type="RadialScaleType" minOccurs="1"
maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
    <xsd:complexType name="LinearScalesType">
        <xsd:sequence>
            <xsd:element name="LinearScale" type="LinearScaleType" minOccurs="1"
maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
    <xsd:complexType name="NumericIndicatorRangesType">
        <xsd:sequence>
            <xsd:element name="NumericIndicatorRange" type="NumericIndicatorRangeType"
minOccurs="1" maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
    <xsd:complexType name="IndicatorStatesType">
        <xsd:sequence>
            <xsd:element name="IndicatorState" type="IndicatorStateType" minOccurs="1"
maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
    <xsd:complexType name="RadialPointersType">
        <xsd:sequence>
            <xsd:element name="RadialPointer" type="RadialPointerType" minOccurs="1"
maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
    <xsd:complexType name="LinearPointersType">
        <xsd:sequence>
            <xsd:element name="LinearPointer" type="LinearPointerType" minOccurs="1"
maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
    <xsd:complexType name="ScaleRangesType">
        <xsd:sequence>
            <xsd:element name="ScaleRange" type="ScaleRangeType" minOccurs="1"
maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>
    <xsd:complexType name="CustomLabelsType">
        <xsd:sequence>
            <xsd:element name="CustomLabel" type="CustomLabelType" minOccurs="1"
maxOccurs="unbounded" />
        </xsd:sequence>
        <xsd:anyAttribute namespace="##other" processContents="lax" />
    </xsd:complexType>

    <xsd:complexType name="MapType">
        <xsd:choice minOccurs="0" maxOccurs="unbounded">
            <!--ReportItemTypeInfo-->
            <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
            <xsd:element name="Top" type="SizeType" minOccurs="0" />
            <xsd:element name="Left" type="SizeType" minOccurs="0" />
        </xsd:choice>
    </xsd:complexType>

```

```

<xsd:element name="Height" type="SizeType" minOccurs="0" />
<xsd:element name="Width" type="SizeType" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
<xsd:element name="Bookmark" type="xsd:string" minOccurs="0" />
<xsd:element name="RepeatWith" type="xsd:string" minOccurs="0" />
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0" />
<!--ReportItemTypeEnd-->
<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
<xsd:element name="PageName" type="xsd:string" minOccurs="0" />
<xsd:element name="Style" type="StyleType" minOccurs="0" />
<xsd:element name="TileLanguage" type="xsd:string" minOccurs="0" />
<xsd:element name="MapLayers" type="MapLayersType" minOccurs="0" />
<xsd:element name="MapDataRegions" type="MapDataRegionsType" minOccurs="0" />
<xsd:element name="MapViewport" type="MapViewportType" minOccurs="1" />
<xsd:element name="MapLegends" type="MapLegendsType" minOccurs="0" />
<xsd:element name="MapTitles" type="MapTitlesType" minOccurs="0" />
<xsd:element name="MapDistanceScale" type="MapDistanceScaleType" minOccurs="0" />
<xsd:element name="MapColorScale" type="MapColorScaleType" minOccurs="0" />
<xsd:element name="MapBorderSkin" type="MapBorderSkinType" minOccurs="0" />
<xsd:element name="AntiAliasing" type="xsd:string" minOccurs="0" />
<xsd:element name="TextAntiAliasingQuality" type="xsd:string" minOccurs="0" />
<xsd:element name="ShadowIntensity" type="xsd:string" minOccurs="0" />
<xsd:element name="MaximumSpatialElementCount" type="xsd:unsignedInt" minOccurs="0" />
<xsd:element name="MaximumTotalPointCount" type="xsd:unsignedInt" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapDataRegionsType">
  <xsd:sequence>
    <xsd:element name="MapDataRegion" type="MapDataRegionType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLayersType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapTileLayer" type="MapTileLayerType" />
    <xsd:element name="MapPolygonLayer" type="MapPolygonLayerType" />
    <xsd:element name="MapPointLayer" type="MapPointLayerType" />
    <xsd:element name="MapLineLayer" type="MapLineLayerType" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLegendsType">
  <xsd:sequence>
    <xsd:element name="MapLegend" type="MapLegendType" minOccurs="1" maxOccurs="unbounded"
/>
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapTitlesType">
  <xsd:sequence>
    <xsd:element name="MapTitle" type="MapTitleType" minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

<xsd:complexType name="MapDataRegionType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="0" />
    <xsd:element name="Filters" type="FiltersType" minOccurs="0" />
    <xsd:element name="MapMember" type="MapMemberType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapMemberType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Group" type="GroupType" minOccurs="1" />
    <xsd:element name="MapMember" type="MapMemberType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapTileLayerType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapLayerStart-->
    <xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <!--MapLayerEnd-->
    <xsd:element name="TileStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="UseSecureConnection" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapTiles" type="MapTilesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapTilesType">
  <xsd:sequence>
    <xsd:element name="MapTile" type="MapTileType" minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapTileType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Name" type="xsd:string" minOccurs="1" />
    <xsd:element name="TileData" type="xsd:string" minOccurs="1" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPolygonLayerType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapLayerStart-->
    <xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <!--MapLayerEnd-->
    <!--MapVectorLayerStart-->
    <xsd:element name="MapDataRegionName" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType" minOccurs="0" />
  </xsd:choice>
  <xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType" minOccurs="0" />
  <xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
</xsd:complexType>

```



```

<xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType" minOccurs="0" />
<xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType" minOccurs="0"
/>
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
<xsd:element name="DataElementOutput" minOccurs="0">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:enumeration value="Output" />
      <xsd:enumeration value="NoOutput" />
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<!--MapVectorLayerEnd-->
<xsd:element name="MapPolygonTemplate" type="MapPolygonTemplateType" minOccurs="0" />
<xsd:element name="MapPolygonRules" type="MapPolygonRulesType" minOccurs="0" />
<xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType" minOccurs="0" />
<xsd:element name="MapCenterPointRules" type="MapPointRulesType" minOccurs="0" />
<xsd:element name="MapPolygons" type="MapPolygonsType" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPointLayerType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapLayerStart-->
    <xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <!--MapLayerEnd-->
    <!--MapVectorLayerStart-->
    <xsd:element name="MapDataRegionName" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType" minOccurs="0"
/>
    <xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType" minOccurs="0" />
    <xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
    <xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType" minOccurs="0" />
    <xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType" minOccurs="0"
/>
  </xsd:choice>
  <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
  <xsd:element name="DataElementOutput" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Output" />
        <xsd:enumeration value="NoOutput" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <!--MapVectorLayerEnd-->
  <xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType" minOccurs="0" />
  <xsd:element name="MapPointRules" type="MapPointRulesType" minOccurs="0" />
  <xsd:element name="MapPoints" type="MapPointsType" minOccurs="0" />
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLineLayerType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapLayerStart-->
    <xsd:element name="VisibilityMode" type="xsd:string" minOccurs="0" />
    <xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="Transparency" type="xsd:string" minOccurs="0" />
    <!--MapLayerEnd-->
    <!--MapVectorLayerStart-->

```

```

    <xsd:element name="MapDataRegionName" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType" minOccurs="0"
  />
  <xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType" minOccurs="0" />
  <xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
  <xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType" minOccurs="0" />
  <xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType" minOccurs="0"
  />
  <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
  <xsd:element name="DataElementOutput" minOccurs="0">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:enumeration value="Output" />
        <xsd:enumeration value="NoOutput" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
  <!--MapVectorLayerEnd-->
  <xsd:element name="MapLineTemplate" type="MapLineTemplateType" minOccurs="0" />
  <xsd:element name="MapLineRules" type="MapLineRulesType" minOccurs="0" />
  <xsd:element name="MapLines" type="MapLinesType" minOccurs="0" />
  <xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapShapefileType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Source" type="xsd:string" minOccurs="1" />
    <xsd:element name="MapFieldNames" type="MapFieldNamesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapSpatialDataSetType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="DataSetName" type="xsd:string" minOccurs="1" />
    <xsd:element name="SpatialField" type="xsd:string" minOccurs="1" />
    <xsd:element name="MapFieldNames" type="MapFieldNamesType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapSpatialDataRegionType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPolygonsType">
  <xsd:sequence>
    <xsd:element name="MapPolygon" type="MapPolygonType" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPointsType">
  <xsd:sequence>
    <xsd:element name="MapPoint" type="MapPointType" minOccurs="1" maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLinesType">

```

```

    <xsd:sequence>
      <xsd:element name="MapLine" type="MapLineType" minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>

  <xsd:complexType name="MapPolygonType">
    <xsd:choice maxOccurs="unbounded">
      <!--MapSpatialElement Start-->
      <xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
      <xsd:element name="MapFields" type="MapFieldsType" minOccurs="0" />
      <!--MapSpatialElement End-->
      <xsd:element name="UseCustomPolygonTemplate" type="xsd:string" minOccurs="0" />
      <xsd:element name="MapPolygonTemplate" type="MapPolygonTemplateType" minOccurs="0" />
      <xsd:element name="UseCustomCenterPointTemplate" type="xsd:string" minOccurs="0" />
      <xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType" minOccurs="0" />
      <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>

  <xsd:complexType name="MapPointType">
    <xsd:choice maxOccurs="unbounded">
      <!--MapSpatialElement Start-->
      <xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
      <xsd:element name="MapFields" type="MapFieldsType" minOccurs="0" />
      <!--MapSpatialElement End-->
      <xsd:element name="UseCustomPointTemplate" type="xsd:string" minOccurs="0" />
      <xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType" minOccurs="0" />
      <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>

  <xsd:complexType name="MapLineType">
    <xsd:choice maxOccurs="unbounded">
      <!--MapSpatialElement Start-->
      <xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
      <xsd:element name="MapFields" type="MapFieldsType" minOccurs="0" />
      <!--MapSpatialElement End-->
      <xsd:element name="UseCustomLineTemplate" type="xsd:string" minOccurs="0" />
      <xsd:element name="MapLineTemplate" type="MapLineTemplateType" minOccurs="0" />
      <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>

  <xsd:complexType name="MapFieldNamesType">
    <xsd:sequence>
      <xsd:element name="MapFieldName" type="xsd:string" minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>

  <xsd:complexType name="MapFieldDefinitionsType">
    <xsd:sequence>
      <xsd:element name="MapFieldDefinition" type="MapFieldDefinitionType" minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>

  <xsd:complexType name="MapFieldDefinitionType">
    <xsd:choice maxOccurs="unbounded">
      <xsd:element name="Name" type="xsd:string" minOccurs="1" />
      <xsd:element name="DataType" minOccurs="1">
        <xsd:simpleType>

```

```

        <xsd:restriction base="xsd:string">
            <xsd:enumeration value="Boolean" />
            <xsd:enumeration value="DateTime" />
            <xsd:enumeration value="Integer" />
            <xsd:enumeration value="Float" />
            <xsd:enumeration value="String" />
            <xsd:enumeration value="Decimal" />
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapFieldsType">
    <xsd:sequence>
        <xsd:element name="MapField" type="MapFieldType" minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapFieldType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="Name" type="xsd:string" minOccurs="1" />
        <xsd:element name="Value" type="xsd:string" minOccurs="1" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapBindingFieldPairsType">
    <xsd:sequence>
        <xsd:element name="MapBindingFieldPair" type="MapBindingFieldPairType" minOccurs="1"
maxOccurs="unbounded" />
    </xsd:sequence>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapBindingFieldPairType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="FieldName" type="xsd:string" minOccurs="1" />
        <xsd:element name="BindingExpression" type="xsd:string" minOccurs="1" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPolygonTemplateType">
    <xsd:choice maxOccurs="unbounded">
        <!--MapSpatialElementTemplateType Start-->
        <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
        <xsd:element name="OffsetX" type="xsd:string" minOccurs="0" />
        <xsd:element name="OffsetY" type="xsd:string" minOccurs="0" />
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="Label" type="xsd:string" minOccurs="0" />
        <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
        <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
        <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
        <xsd:element name="DataElementOutput" minOccurs="0">
            <xsd:simpleType>
                <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="Output" />
                    <xsd:enumeration value="NoOutput" />
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:element>
        <xsd:element name="DataElementLabel" type="xsd:string" minOccurs="0" />
        <!--MapSpatialElementTemplateType End-->
    </xsd:choice>

```

```

    <xsd:element name="ScaleFactor" type="xsd:string" minOccurs="0" />
    <xsd:element name="CenterPointOffsetX" type="xsd:string" minOccurs="0" />
    <xsd:element name="CenterPointOffsetY" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowLabel" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapMarkerTemplateType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSpatialElementTemplateType Start-->
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetY" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementLabel" type="xsd:string" minOccurs="0" />
    <!--MapSpatialElementTemplateType End-->
    <!--MapPointTemplateType Start-->
    <xsd:element name="Size" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
    <!--MapPointTemplateTypeType End-->
    <xsd:element name="MapMarker" type="MapMarkerType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLineTemplateType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSpatialElementTemplateType Start-->
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetX" type="xsd:string" minOccurs="0" />
    <xsd:element name="OffsetY" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="Label" type="xsd:string" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element name="DataElementLabel" type="xsd:string" minOccurs="0" />
    <!--MapSpatialElementTemplateType End-->
    <xsd:element name="Width" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

<xsd:complexType name="MapBucketsType">
  <xsd:sequence>
    <xsd:element name="MapBucket" type="MapBucketType" minOccurs="1" maxOccurs="unbounded"
  />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapBucketType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapColorRangeRuleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapAppearanceRule Start-->
    <xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapAppearanceRule End-->
    <!--MapColorRule Start-->
    <xsd:element name="ShowInColorScale" type="xsd:string" minOccurs="0" />
    <!--MapColorRule End-->
    <xsd:element name="StartColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="MiddleColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapColorPaletteRuleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapAppearanceRule Start-->
    <xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapAppearanceRule End-->
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

    </xsd:element>
    <!--MapAppearanceRule End-->
    <!--MapColorRule Start-->
    <xsd:element name="ShowInColorScale" type="xsd:string" minOccurs="0" />
    <!--MapColorRule End-->
    <xsd:element name="Palette" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapCustomColorRuleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapAppearanceRule Start-->
    <xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapAppearanceRule End-->
    <!--MapColorRule Start-->
    <xsd:element name="ShowInColorScale" type="xsd:string" minOccurs="0" />
    <!--MapColorRule End-->
    <xsd:element name="MapCustomColors" type="MapCustomColorsType" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapCustomColorsType">
  <xsd:sequence>
    <xsd:element name="MapCustomColor" type="xsd:string" minOccurs="1"
maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapSizeRuleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapAppearanceRule Start-->
    <xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

    </xsd:element>
    <!--MapAppearanceRule End-->
    <xsd:element name="StartSize" type="xsd:string" minOccurs="1" />
    <xsd:element name="EndSize" type="xsd:string" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapMarkerRuleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapAppearanceRule Start-->
    <xsd:element name="DataValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="DistributionType" type="xsd:string" minOccurs="0" />
    <xsd:element name="BucketCount" type="xsd:string" minOccurs="0" />
    <xsd:element name="StartValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="EndValue" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
    <xsd:element name="LegendName" type="xsd:string" minOccurs="0" />
    <xsd:element name="LegendText" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
    <xsd:element name="DataElementOutput" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="Output" />
          <xsd:enumeration value="NoOutput" />
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <!--MapAppearanceRule End-->
    <xsd:element name="MapMarkers" type="MapMarkersType" minOccurs="1" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapMarkersType">
  <xsd:sequence>
    <xsd:element name="MapMarker" type="MapMarkerType" minOccurs="1" maxOccurs="unbounded"
  />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapMarkerType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapMarkerStyle" type="xsd:string" minOccurs="0" />
    <xsd:element name="MapMarkerImage" type="MapMarkerImageType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapMarkerImageType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Source" type="xsd:string" minOccurs="0" />
    <xsd:element name="Value" type="xsd:string" minOccurs="0" />
    <xsd:element name="MIMEType" type="xsd:string" minOccurs="0" />
    <xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ResizeMode" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPolygonRulesType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType" minOccurs="0" />
    <xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType" minOccurs="0" />

```



```

        <xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapPointRulesType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType" minOccurs="0" />
        <xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType" minOccurs="0" />
        <xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType" minOccurs="0" />
        <xsd:element name="MapSizeRule" type="MapSizeRuleType" minOccurs="0" />
        <xsd:element name="MapMarkerRule" type="MapMarkerRuleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLineRulesType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType" minOccurs="0" />
        <xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType" minOccurs="0" />
        <xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType" minOccurs="0" />
        <xsd:element name="MapSizeRule" type="MapSizeRuleType" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapViewportType">
    <xsd:choice maxOccurs="unbounded">
        <!--MapSubItem Start-->
        <xsd:element name="Style" type="StyleType" minOccurs="0" />
        <xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
        <xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
        <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
        <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
        <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
        <xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
        <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
        <!--MapSubItem End-->
        <xsd:element name="MapCoordinateSystem" type="xsd:string" minOccurs="0" />
        <xsd:element name="MapProjection" type="xsd:string" minOccurs="0" />
        <xsd:element name="ProjectionCenterX" type="xsd:string" minOccurs="0" />
        <xsd:element name="ProjectionCenterY" type="xsd:string" minOccurs="0" />
        <xsd:element name="MapCustomView" type="MapCustomViewType" minOccurs="0" />
        <xsd:element name="MapElementView" type="MapElementViewType" minOccurs="0" />
        <xsd:element name="MapDataBoundView" type="MapDataBoundViewType" minOccurs="0" />
        <xsd:element name="MapLimits" type="MapLimitsType" minOccurs="0" />
        <xsd:element name="MaximumZoom" type="xsd:string" minOccurs="0" />
        <xsd:element name="MinimumZoom" type="xsd:string" minOccurs="0" />
        <xsd:element name="SimplificationResolution" type="xsd:string" minOccurs="0" />
        <xsd:element name="ContentMargin" type="xsd:string" minOccurs="0" />
        <xsd:element name="MapMeridians" type="MapGridLinesType" minOccurs="0" />
        <xsd:element name="MapParallels" type="MapGridLinesType" minOccurs="0" />
        <xsd:element name="GridUnderContent" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLimitsType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="MinimumX" type="xsd:string" minOccurs="0" />
        <xsd:element name="MinimumY" type="xsd:string" minOccurs="0" />
        <xsd:element name="MaximumX" type="xsd:string" minOccurs="0" />
        <xsd:element name="MaximumY" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>

```

```

    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapCustomViewType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Zoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="CenterX" type="xsd:string" minOccurs="0" />
    <xsd:element name="CenterY" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapElementViewType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Zoom" type="xsd:string" minOccurs="0" />
    <xsd:element name="LayerName" type="xsd:string" minOccurs="1" />
    <xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType" minOccurs="0"
/>
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapDataBoundViewType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Zoom" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapDistanceScaleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSubItem Start-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
    <xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <!--MapSubItem End-->
    <!--MapDockableSubItem Start-->
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <!--MapDockableSubItem End-->
    <xsd:element name="ScaleColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="ScaleBorderColor" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapColorScaleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSubItem Start-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
    <xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

```

```

<!--MapSubItem End-->
<!--MapDockableSubItem Start-->
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
<xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<!--MapDockableSubItem End-->
<xsd:element name="MapColorScaleTitle" type="MapColorScaleTitleType" minOccurs="0" />
<xsd:element name="TickMarkLength" type="xsd:string" minOccurs="0" />
<xsd:element name="ColorBarBorderColor" type="xsd:string" minOccurs="0" />
<xsd:element name="LabelInterval" type="xsd:string" minOccurs="0" />
<xsd:element name="LabelFormat" type="xsd:string" minOccurs="0" />
<xsd:element name="LabelPlacement" type="xsd:string" minOccurs="0" />
<xsd:element name="LabelBehavior" type="xsd:string" minOccurs="0" />
<xsd:element name="HideEndLabels" type="xsd:string" minOccurs="0" />
<xsd:element name="RangeGapColor" type="xsd:string" minOccurs="0" />
<xsd:element name="NoDataText" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapColorScaleTitleType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Caption" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapTitleType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSubItem Start-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
    <xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
    <!--MapSubItem End-->
    <!--MapDockableSubItem Start-->
    <xsd:element name="Position" type="xsd:string" minOccurs="0" />
    <xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
    <xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
    <!--MapDockableSubItem End-->
    <xsd:element name="Text" type="xsd:string" minOccurs="0" />
    <xsd:element name="Angle" type="xsd:string" minOccurs="0" />
    <xsd:element name="TextShadowOffset" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLegendType">
  <xsd:choice maxOccurs="unbounded">
    <!--MapSubItem Start-->
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:element name="MapLocation" type="MapLocationType" minOccurs="0" />
    <xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
    <xsd:element name="LeftMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="RightMargin" type="xsd:string" minOccurs="0" />
    <xsd:element name="TopMargin" type="xsd:string" minOccurs="0" />
  </xsd:choice>

```

```

<xsd:element name="BottomMargin" type="xsd:string" minOccurs="0" />
<xsd:element name="ZIndex" type="xsd:string" minOccurs="0" />
<!--MapSubItem End-->
<!--MapDockableSubItem Start-->
<xsd:element name="Position" type="xsd:string" minOccurs="0" />
<xsd:element name="DockOutsideViewport" type="xsd:string" minOccurs="0" />
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
<xsd:element name="ToolTip" type="xsd:string" minOccurs="0" />
<!--MapDockableSubItem End-->
<xsd:element name="Layout" type="xsd:string" minOccurs="0" />
<xsd:element name="MapLegendTitle" type="MapLegendTitleType" minOccurs="0" />
<xsd:element name="AutoFitTextDisabled" type="xsd:string" minOccurs="0" />
<xsd:element name="MinFontSize" type="xsd:string" minOccurs="0" />
<xsd:element name="InterlacedRows" type="xsd:string" minOccurs="0" />
<xsd:element name="InterlacedRowsColor" type="xsd:string" minOccurs="0" />
<xsd:element name="EquallySpacedItems" type="xsd:string" minOccurs="0" />
<xsd:element name="TextWrapThreshold" type="xsd:string" minOccurs="0" />
<xsd:any namespace="##other" processContents="lax" />
</xsd:choice>
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
<xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLegendTitleType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Caption" type="xsd:string" minOccurs="0" />
    <xsd:element name="TitleSeparator" type="xsd:string" minOccurs="0" />
    <xsd:element name="TitleSeparatorColor" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapGridLinesType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
    <xsd:element name="Interval" type="xsd:string" minOccurs="0" />
    <xsd:element name="ShowLabels" type="xsd:string" minOccurs="0" />
    <xsd:element name="LabelPosition" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapBorderSkinType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="MapBorderSkinType" type="xsd:string" minOccurs="0" />
    <xsd:element name="Style" type="StyleType" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapLocationType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Left" type="xsd:string" minOccurs="0" />
    <xsd:element name="Top" type="xsd:string" minOccurs="0" />
    <xsd:element name="Unit" type="xsd:string" minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="MapSizeType">
  <xsd:choice maxOccurs="unbounded">
    <xsd:element name="Width" type="xsd:string" minOccurs="1" />

```

```

        <xsd:element name="Height" type="xsd:string" minOccurs="1" />
        <xsd:element name="Unit" type="xsd:string" minOccurs="0" />
        <xsd:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>

<xsd:complexType name="ReportParametersLayoutType">
    <xsd:choice>
        <xsd:element name="GridLayoutDefinition" type="GridLayoutDefinitionType" minOccurs="1"
/>
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="GridLayoutDefinitionType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="NumberOfColumns" type="xsd:int" minOccurs="1" />
        <xsd:element name="NumberOfRows" type="xsd:int" minOccurs="1" />
        <xsd:element name="CellDefinitions" type="CellDefinitionsType" minOccurs="0" />
    </xsd:choice>
</xsd:complexType>

<xsd:complexType name="CellDefinitionsType">
    <xsd:sequence>
        <xsd:element name="CellDefinition" type="CellDefinitionType" minOccurs="1"
maxOccurs="unbounded" />
    </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="CellDefinitionType">
    <xsd:choice maxOccurs="unbounded">
        <xsd:element name="ColumnIndex" type="xsd:int" minOccurs="1" />
        <xsd:element name="RowIndex" type="xsd:int" minOccurs="1" />
        <xsd:element name="ParameterName" type="xsd:normalizedString" minOccurs="1" />
    </xsd:choice>
</xsd:complexType>
</xsd:schema>

```

## 6 Appendix B: Error Codes

### 6.1 rsAggregateInFilterExpression

*rsAggregateInFilterExpression* is a critical error that will occur when one of the **aggregate functions** other than [Previous](#), [RunningValue](#), or [RowNumber](#) is specified within a [Filter.FilterExpression](#) or [Filter.FilterValues](#) instance.

### 6.2 rsAggregateInGroupExpression

*rsAggregateInGroupExpression* is a critical error that will occur when one of the **aggregate functions** other than [Previous](#) or [RunningValue](#) is specified within a [GroupExpressions.GroupExpression](#) instance.

### 6.3 rsAggregateInQueryParameterExpression

*rsAggregateInQueryParameterExpression* is a critical error that will occur when one of the **aggregate functions** other than [Previous](#), [RunningValue](#), or [RowNumber](#) or a [Lookup](#), [LookupSet](#), or [MultiLookup](#) function is specified within a [QueryParameter.Value](#) instance.

### 6.4 rsAggregateInReportParameterExpression

*rsAggregateInReportParameterExpression* is a critical error that will occur when one of the aggregate functions other than [Previous](#), [RunningValue](#), or [RowNumber](#) or a [Lookup](#), [LookupSet](#), or [MultiLookup](#) function is specified within any expression in a [ReportParameter](#) instance.

### 6.5 rsAggregateInReportLanguageExpression

*rsAggregateInReportLanguageExpression* is a critical error that will occur when one of the aggregate functions other than [Previous](#), [RunningValue](#), or [RowNumber](#) or a [Lookup](#), [LookupSet](#), or [MultiLookup](#) function is specified within any expression in a [Report.Language](#) instance.

### 6.6 rsAggregateInCalculatedFieldExpression

*rsAggregateInCalculatedFieldExpression* is a critical error that will occur when one of the aggregate functions, or a [Lookup](#), [LookupSet](#), or [MultiLookup](#) function is specified within any expression within a [Field](#) instance.

### 6.7 rsAggregateofAggregate

*rsAggregateofAggregate* is a critical error that will occur when one of the aggregate functions is specified as an argument to another aggregate function.

### 6.8 rsAggregateReportItemInBody

*rsAggregateReportItemInBody* is a critical error that occurs when one of the aggregate functions has an argument referencing a [ReportItems](#) **global collection**.

### 6.9 rsBinaryConstant

*rsBinaryConstant* is a critical error that will occur when an element expects a binary value, such as an [Image.Value](#) instance when [Image.Source](#) is set to "Database", and gets a constant value instead.

## 6.10 rsChartSeriesPlotTypeIgnored

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsChartSeriesPlotTypeIgnored* is a non-critical error that will occur when a [ChartSeries.PlotType](#) element is ignored because the corresponding [Chart.Type](#) is not set to "Column".

## 6.11 rsCompilerErrorInExpression

*rsCompilerErrorInExpression* is a critical error that will occur whenever the evaluation of an expression results in a compile error. For example, if an expression is set to "=myFunction()", where "myFunction()" is not a declared or recognizable function, this error will occur.

## 6.12 rsCompilerErrorInCode

*rsCompilerErrorInCode* is a critical error that will occur whenever a [Report.Code](#) element contains a compile error. For example, if the first line in a **Report.Code** element is set to "myFunction()", where "myFunction()" is not a declared or recognizable function, this error will occur.

## 6.13 rsCompilerErrorInClassInstanceDeclaration

*rsCompilerErrorInClassInstanceDeclaration* is a critical error that occurs whenever a compile error occurs in a [Class](#) instance. For example, if [Class.ClassName](#) is set to "Math", but the **System.Math** class has no constructors, this error will occur.

## 6.14 rsUnexpectedCompilerError

*rsUnexpectedCompilerError* is a critical error that will occur when an expression evaluation returns an unexpected compiler error. The specific compiler error is included as an inner error message.

## 6.15 rsConflictingRunningValueScopesInMatrix

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsConflictingRunningValueScopesInMatrix* is a critical error that occurs when an expression within a [Matrix](#) contains both a [Previous](#) aggregate function that is associated with a **Matrix** row scope and a **Previous** aggregate function that is associated with a **Matrix** column scope.

This critical error occurs when an expression within a **Matrix** contains both a [RowNumber](#) aggregate function that is associated with a **Matrix** row scope and a **RowNumber** aggregate function that is associated with a **Matrix** column scope.

This critical error occurs when an expression within a **Matrix** contains both a [RunningValue](#) aggregate function that is associated with a **Matrix** row scope and a **RunningValue** aggregate function that is associated with a **Matrix** column scope.

## 6.16 rsConflictingRunningValueScopesInTablix

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsConflictingRunningValueScopesInTablix* is a critical error that occurs when an expression within a [Tablix](#) contains both a [Previous](#) aggregate function that is associated with a **Tablix** row scope and a **Previous** aggregate function that is associated with a **Tablix** column scope.

This critical error occurs when an expression within a **Tablix** contains both a [RunningValue](#) aggregate function that is associated with a **Tablix** row scope and a **RunningValue** aggregate function that is associated with a **Tablix** column scope.

This critical error occurs when an expression within a **Tablix** contains both a [RowNumber](#) aggregate function that is associated with a **Tablix** row scope and a **RowNumber** aggregate function that is associated with a **Tablix** column scope.

## 6.17 rsCountRowsInPageSectionExpression

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsCountRowsInPageSectionExpression* is a critical error that will occur (in Microsoft SQL Server 2005 Reporting Services) when the [CountRows](#) aggregate function is used in an expression within a [PageHeaderFooter](#).

## 6.18 rsCountStarNotSupported

*rsCountStarNotSupported* is a critical error that will occur whenever an expression contains any form of the function **Count(\*, scope)**.

## 6.19 rsCountStarRVNotSupported

*rsCountStarRVNotSupported* is a critical error that will occur whenever an expression contains any form of the function **RunningValue(\*, Count, scope)**.

## 6.20 rsCustomAggregateAndFilter

*rsCustomAggregateAndFilter* is a critical error that will occur whenever the [Aggregate](#) function is used in a report that contains a [Filter](#) instance.

## 6.21 rsDataRegionInDetailList

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsDataRegionInDetailList* is a critical error that will occur when a [Table](#), [List](#), [Matrix](#), [Chart](#), or [Subreport](#) element exists within a **List** element that does not have [List.Grouping](#) specified.

## 6.22 rsDataRegionInPageSection

*rsDataRegionInPageSection* is a critical error that will occur whenever a [Tablix](#), [Chart](#), [GaugePanel](#) or [Subreport](#) element exists within either a [Page.PageHeader](#) element or a [Page.PageFooter](#) element, or when a [Table](#), [Chart](#), [Subreport](#), [List](#), or [Matrix](#) element exists within either a [PageSection](#) element or a [PageHeaderFooter](#) element.

## 6.23 rsDataRegionInTableDetailRow

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsDataRegionInTableDetailRow* is a critical error that will occur when a [Table](#), [List](#), [Matrix](#), [SubReport](#), or [Chart](#) exists inside of a [TableRow](#) within a [Details.TableRows](#) element.

## 6.24 rsDataRegionWithoutDataSet

*rsDataRegionWithoutDataSet* is a critical error that will occur whenever a [Tablix](#), [GaugePanel](#), [Chart](#), or [Subreport](#) instance occurs within a [Report](#) that does not have an associated [DataSet](#).



## 6.25 rsDataSourceReferenceNotPublished

*rsDataSourceReferenceNotPublished* is a non-critical error that will occur when a [DataSource.DataSourceReference](#) element refers to the name of a DataSource that does not exist.

## 6.26 rsDuplicateChartAxisName

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsDuplicateChartAxisName* is a critical error that will occur when there are two or more [ChartAxis](#) elements with the same **Name** attribute within the same [ChartArea](#) element.

## 6.27 rsSpecifiedNonValueAxisName

*rsSpecifiedNonValueAxisName* is a critical error that will occur when a [ChartSeries.ValueAxisName](#) element contains a value that matches the **Name** attribute of a [ChartCategoryAxes.ChartAxis](#) element instead of a **ChartValueAxes.ChartAxis** element for the same [ChartArea](#) element.

## 6.28 rsValueAxisNameNotFound

*rsValueAxisNameNotFound* is a critical error that will occur when a [ChartSeries.ValueAxisName](#) element contains a value that does not match the **Name** attribute of any [ChartAxis](#) element within a [ChartArea](#) element.

## 6.29 rsInvalidTextEffect

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidTextEffect* is a critical error that will occur when the [Style.TextEffect](#) element is set to a constant [String](#) ([XMLSCHEMA2/2](#) section 3.2.1) that is not one of its valid values.

## 6.30 rsInvalidBackgroundHatchType

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidBackgroundHatchType* is a critical error that will occur when a [Style.BackgroundHatchType](#) element is set to a constant **String** that is not one of its valid values.

## 6.31 rsInvalidBackgroundImagePosition

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidBackgroundImagePosition* is a critical error that will occur when a [BackgroundImage.Position](#) element is not set to one of its valid values.

## 6.32 rsPageBreakOnGaugeGroup

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsPageBreakOnGaugeGroup* is a non-critical error that will occur when a [GaugePanel.PageBreak](#) element is set to create a page break for each [Group](#) within the [GaugePanel](#).

## 6.33 rsDuplicateChartLegendCustomItemCellName

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsDuplicateChartLegendCustomItemCellName* is a critical error that will occur when two or more [ChartLegendCustomItemCell](#) elements have the same **Name** attribute within the same [ChartLegendCustomItemCells](#) collection.

### 6.34 rsDuplicateChartFormulaParameter

*Applies to* [RDL 2008/01](#)

*rsDuplicateChartFormulaParameter* is a critical error that will occur when two or more [ChartFormulaParameters.ChartFormulaParameter](#) elements have the same **Name** attribute within the same [ChartFormulaParameters](#) collection.

### 6.35 rsDuplicateClassInstanceName

*rsDuplicateClassInstanceName* is a critical error that will occur when two or more [Class.InstanceName](#) instances have the same value.

### 6.36 rsDuplicateDataSourceName

*rsDuplicateDataSourceName* is a critical error that will occur when two or more [DataSource](#) instances have the same **Name** attribute values.

### 6.37 rsInvalidDataSourceNameLength

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidDataSourceNameLength* is a critical error that will occur whenever the value of the **Name** attribute of the [DataSource](#) is empty or is longer than 256 characters.

### 6.38 rsDuplicateEmbeddedImageName

*rsDuplicateEmbeddedImageName* is a critical error that will occur whenever two or more [EmbeddedImage](#) elements have the same **Name** attribute within the same [Report](#).

### 6.39 rsInvalidEmbeddedImageNameNotCLSCompliant

*rsInvalidEmbeddedImageNameNotCLSCompliant* is a critical error that will occur whenever the value of the **Name** attribute of an [EmbeddedImage](#) element is not CLS-compliant [[UTR15](#)].

### 6.40 rsInvalidEmbeddedImageNameLength

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidEmbeddedImageNameLength* is a critical error that will occur whenever the **Name** attribute of an [EmbeddedImage](#) element has a value that is empty or greater than 256 characters in length.

### 6.41 rsDuplicateFieldName

*rsDuplicateFieldName* is a critical error that will occur whenever two or more [Field](#) elements within the same [DataSet](#) element contain the same **Name** attribute.

### 6.42 rsDuplicateReportItemName

*rsDuplicateReportItemName* is a critical error that occurs whenever two or more report item elements within a [ReportItems](#) collection contain the same **Name** attribute.

### 6.43 rsDuplicateReportParameterName

*rsDuplicateReportParameterName* is a critical error that will occur whenever two or more [ReportParameter](#) elements contain the same name.

### 6.44 rsDuplicateCaseInsensitiveReportParameterName

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsDuplicateCaseInsensitiveReportParameterName* is a non-critical error that will occur whenever two or more [ReportParameter](#) elements contain names that differ, but only in terms of case.

### 6.45 rsDuplicateScopeName

*rsDuplicateScopeName* is a critical error that will occur whenever two or more [Group](#) instances, data region elements, or [DataSet](#) elements contain the same name within a [Report](#).

### 6.46 rsExpressionMissingCloseParen

*rsExpressionMissingCloseParen* is a critical error that will occur whenever an expression contains a function call that contains an open parenthesis character (()) but does not contain a matching close parenthesis character.

### 6.47 rsFieldInPageSectionExpression

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsFieldInPageSectionExpression* is a critical error that will occur if an expression within a [PageSection](#), [PageHeaderFooter](#), or [PageFooter](#) element contains a reference to a [Field](#) via the [Fields global collection](#).

### 6.48 rsFieldInReportParameterExpression

*rsFieldInReportParameterExpression* is a critical error that will occur whenever a [Field](#) via the [Fields global collection](#) in an expression within a [ReportParameter](#).

### 6.49 rsFieldInQueryParameterExpression

*rsFieldInQueryParameterExpression* is a critical error that will occur whenever a [Field](#) is referenced via the [Fields global collection](#) in an expression within a [QueryParameter](#).

### 6.50 rsFieldInReportLanguageExpression

*rsFieldInReportLanguageExpression* is a critical error that will occur whenever a [Field](#) is referenced via the [Fields global collection](#) in an expression within a [Report.Language](#) instance.

### 6.51 rsGlobalNotDefined

*rsGlobalNotDefined* is a critical error that will occur whenever an expression references "Aggregates" as a member of the [Globals](#) collection.

### 6.52 rsInvalidAction

*rsInvalidAction* is a critical error that will occur when an [Action](#) item contains more than one of the following elements: [Action.BookmarkLink](#), [Action.Hyperlink](#), or [Action.Drillthrough](#).

## 6.53 rsInvalidAggregateScope

*rsInvalidAggregateScope* is a critical error that occurs whenever one of the aggregate functions contains a scope argument that is not defined or valid.

## 6.54 rsInvalidAltReportItem

*rsInvalidAltReportItem* is a critical error that will occur when a [CustomReportItem.AltReportItem](#) contains another [CustomReportItem](#) element.

## 6.55 rsInvalidBooleanConstant

*rsInvalidBooleanConstant* is a critical error that will occur when an element that requires a **Boolean** value or the value of a **Boolean** expression is not set to one of the constant values "1", "0", "True" (with uppercase T), or "False" (with uppercase F).

## 6.56 rsInvalidCategoryGrouping

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsInvalidCategoryGrouping* is a critical error that will occur when a [CategoryGrouping](#) element contains either none or both of the following elements: [CategoryGrouping.DynamicCategories](#) and [CategoryGrouping.StaticCategories](#).

## 6.57 rsInvalidCharacterInExpression

*rsInvalidCharacterInExpression* is a critical error that will occur when an expression contains either a line terminator or a colon character.

## 6.58 rsInvalidChartGroupings

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsInvalidChartGroupings* is a critical error that will occur if a [Chart](#) that has [Chart.Type](#) set to "Pie" or "Doughnut" contains both [SeriesGrouping.StaticSeries](#) elements and [CategoryGrouping.StaticCategories](#) elements.

## 6.59 rsInvalidChartSubType

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsInvalidChartSubType* is a critical error that will occur if, for a [Chart](#), the [Chart.Subtype](#) value is not compatible with the [Chart.Type](#).

## 6.60 rsInvalidColumnGrouping

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsInvalidColumnGrouping* is a critical error that will occur (if a [ColumnGrouping](#) contains either none or both of the following elements: [ColumnGrouping.DynamicColumns](#) and [ColumnGrouping.StaticColumns](#)).

## 6.61 rsInvalidCustomAggregateExpression

*rsInvalidCustomAggregateExpression* is a critical error that occurs when the [Aggregate](#) function does not contain as its *Expression* parameter a simple [Fields global collection](#).

## 6.62 rsInvalidCustomAggregateScope

*rsInvalidCustomAggregateScope* is a critical error that occurs when the [Aggregate](#) function does not have as its *Scope* parameter the name of a scope where all containing [Group](#) scopes have group expressions that are either simple field references or constants.

## 6.63 rsInvalidCustomPropertyName

*rsInvalidCustomPropertyName* is a critical error that will occur when one of the following conditions is true:

- Two or more [CustomProperty](#) elements contain the same [CustomProperty.Name](#) value within a [CustomProperties](#) collection.
- A **CustomProperty.Name** is empty or is not specified.

## 6.64 rsInvalidDataElementNameNotCLSCompliant

*rsInvalidDataElementNameNotCLSCompliant* is a critical error that will occur when any element with a **DataElementName** child element (such as [Tablix.DataElementName](#)) contains a value that is not CLS-compliant [[UTR15](#)].

## 6.65 rsInvalidDataSetName

*rsInvalidDataSetName* is a critical error that will occur when any element that has a **DataSetName** child element (such as [Tablix.DataSetName](#)) contains a value that refers to a non-existent [DataSet](#).

## 6.66 rsInvalidDataSource

*rsInvalidDataSource* is a critical error that will occur when a [DataSource](#) contains either both or neither of the following elements: [DataSource.DataSourceReference](#) and [DataSource.ConnectionProperties](#).

## 6.67 rsInvalidDataSourceReference

*rsInvalidDataSourceReference* is a critical error that will occur when a [DataSet](#) contains a [Query.DataSourceName](#) which references a non-existent [DataSource](#).

## 6.68 rsInvalidValidValuesDataSetReference

*rsInvalidValidValuesDataSetReference* is a critical error that will occur when a [DataSetReference.DataSetName](#) element within a [ReportParameter.ValidValues](#) element has a value that does not match the name of any [DataSet](#).

## 6.69 rsInvalidDefaultValueDataSetReference

*rsInvalidDefaultValueDataSetReference* is a critical error that will occur when a [DataSetReference.DataSetName](#) element within a [ReportParameter.DefaultValue](#) element has a value that does not match the name of any [DataSet](#).

## 6.70 rsInvalidDataSetReferenceField

*rsInvalidDataSetReferenceField* is a critical error that will occur when a [ReportParameter](#) contains a [DataSetReference](#) that references an existing [DataSet](#), but the peer element [DataSetReference.ValueField](#) references a non-existing [Field](#) within that **DataSet**.

## 6.71 rsInvalidDefaultValue

*rsInvalidDefaultValue* is a critical error that will occur when a [ReportParameter.DefaultValue](#) element contains either none or both of the following elements: [DefaultValue.Values](#) or [DefaultValue.DataSetReference](#).

## 6.72 rsInvalidDefaultValueValues

*rsInvalidDefaultValueValues* is a non-critical error that will occur when a [ReportParameter.DefaultValue](#) contains a [Values](#) collection that has more than one [Values.Value](#) element. For single-valued parameters, only the first **Values.Value** element in the **Values** collection is used.

## 6.73 rsInvalidDetailsDataGrouping

*rsInvalidDetailsDataGrouping* is a critical error that will occur if a details data region member contains an inner data region member. For example, this error will occur if a [TablixMember](#) has no [TablixMember.Group](#) element specified, and it contains a [TablixMember.TablixMembers](#) child element.

## 6.74 rsInvalidExpressionScope

*rsInvalidExpressionScope* is a critical error that occurs if, for a [UserSort](#) element, the [UserSort.SortExpressionScope](#) element specifies a scope that refers to a scope that is not the current scope or contained by the current scope.

## 6.75 rsInvalidExpressionScopeDataSet

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidExpressionScopeDataSet* is a critical error that occurs if, for a [UserSort](#) element, the [UserSort.SortExpressionScope](#) element and the [UserSort.SortTarget](#) element refer to scope data regions that do not share the same [DataSet](#).

## 6.76 rsInvalidSortExpressionScope

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidSortExpressionScope* is a critical error that occurs if a [UserSort.SortExpressionScope](#) is set to a **detail scope**.

## 6.77 rsIneffectiveSortExpressionScope

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsIneffectiveSortExpressionScope* is a non-critical error that occurs if a [Textbox](#) element, with the [Textbox.UserSort](#) element present, specifies the current scope for the [UserSort.SortExpressionScope](#) element of the **Textbox**.

## 6.78 rsInvalidField

*rsInvalidField* is a critical error that will occur when a [Field](#) element contains neither or both of the following elements:

- [Field.DataField](#)
- [Field.Value](#)

## 6.79 rsInvalidFieldNameNotCLSCompliant

*rsInvalidFieldNameNotCLSCompliant* is a critical error that will occur whenever a [Field](#) element has a **Name** attribute that is not CLS-compliant [\[UTR15\]](#).

## 6.80 rsInvalidFieldNameLength

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsInvalidFieldNameLength* is a critical error that will occur when the value of the **Name** attribute of a [Field](#) element is either empty or is longer than 256 characters.

## 6.81 rsInvalidGroupExpressionScope

*rsInvalidGroupExpressionScope* is a critical error that will occur when a [RowNumber](#) aggregate function within a [GroupExpressions.GroupExpression](#) element has its *Scope* parameter set to a value that does not match the name of the [Group](#) that contains the current **Group**.

## 6.82 rsInvalidGroupNameNotCLSCompliant

*rsInvalidGroupNameNotCLSCompliant* is a critical error that will occur when a [Group](#) element has a **Name** attribute whose value is not CLS-compliant [\[UTR15\]](#).

## 6.83 rsInvalidHideDuplicateScope

*rsInvalidHideDuplicateScope* is a critical error that will occur when a [Textbox.HideDuplicates](#) element contains a value that is not the name of its associated [DataSet](#) or the name of a [Group](#) within the current **Group**.

## 6.84 rsInvalidURLProtocol

*rsInvalidURLProtocol* is a critical error that will occur when an element that requires an [RdlURL](#) value does not begin with one of the following valid protocol strings: "http://", "https://", "ftp://", "mailto:", or "news:".

## 6.85 rsInvalidIntegerConstant

*rsInvalidIntegerConstant* is a critical error that occurs when an element that requires an [Integer](#) ([\[XMLSCHEMA2/2\]](#) section 3.3.17) constant (such as [Page.Columns](#)) is not set to an integer constant.

## 6.86 rsInvalidDateTimeConstant

*rsInvalidDateTimeConstant* is a critical error that occurs when an element that requires a [DateTime](#) ([\[XMLSCHEMA2\]](#) section 3.2.7) constant is not set to a **Date**Time**** constant.

## 6.87 rsInvalidFloatConstant

*rsInvalidFloatConstant* is a critical error that occurs when an element that requires a [Float](#) ([\[XMLSCHEMA2\]](#) section 3.2.4) constant is not set to a **Float** constant.

## 6.88 rsLabelExpressionOnChartScalarAxisIsIgnored

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsLabelExpressionOnChartScalarAxisIsIgnored* is a non-critical error that will occur when the [Axis.Scalar](#) element for the [Chart.CategoryAxis](#) has a value of true and the only [Grouping](#) element in the [Chart.CategoryGroupings](#) collection specifies [Grouping.Label](#).

## 6.89 rsInvalidMatrixSubtotalReportItem

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsInvalidMatrixSubtotalReportItem* is a critical error that will occur if a [Subtotal.ReportItems](#) element contains more than one [Textbox](#) element.

## 6.90 rsInvalidNameNotCLSCompliant

*rsInvalidNameNotCLSCompliant* is a critical error that will occur when the value of the **Name** attribute for an element is not CLS-compliant [\[UTR15\]](#).

## 6.91 rsInvalidNameLength

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsInvalidNameLength* is a critical error that will occur when the value of the **Name** attribute for an element is empty or is longer than 256 characters.

## 6.92 rsInvalidNumberOfFilterValues

*rsInvalidNumberOfFilterValues* is a critical error that will occur when a [Filter](#) element contains an incorrect number of [FilterValues.FilterValue](#) elements for a particular [Filter.Operator](#) value. For example, if **Filter.Operator** is set to "Equal", and there exist two **FilterValues.FilterValue** instances for that **Filter**, this error will occur.

## 6.93 rsInvalidOmittedExpressionScope

*rsInvalidOmittedExpressionScope* is a critical error that occurs if, for a [UserSort](#) element, the [UserSort.SortExpressionScope](#) element is omitted and the **UserSort** is in a **detail scope**.

## 6.94 rsInvalidOmittedTargetScope

*rsInvalidOmittedTargetScope* is a non-critical error that will occur if a [Textbox](#) element is not within any data regions, and the [Textbox.UserSort](#) element is present and the [UserSort.SortTarget](#) element is omitted.

## 6.95 rsInvalidParameterNameNotCLSCompliant

*rsInvalidParameterNameNotCLSCompliant* is a critical error that will occur when a [Subreport](#) contains a [Parameters.Parameter](#) element that has a **Name** attribute whose value is not CLS-compliant [\[UTR15\]](#).



## 6.96 rsInvalidParameterNameLength

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsInvalidParameterNameLength* is a critical error that will occur when a [Subreport](#) contains a [Parameters.Parameter](#) element that has a **Name** attribute whose value is empty or is longer than 256 characters.

## 6.97 rsInvalidPreviousAggregateInMatrixCell

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsInvalidPreviousAggregateInMatrixCell* is a critical error that will occur when a [Previous](#) aggregate functions is used in an expression within a [MatrixCell](#) element.

## 6.98 rsInvalidPreviousAggregateInTablixCell

*rsInvalidPreviousAggregateInTablixCell* is a critical error that will occur when a [Previous](#) aggregate functions is used in an expression within a [TablixCell](#) element.

## 6.99 rsInvalidRepeatWith

*rsInvalidRepeatWith* is a critical error that will occur when a data region, [Subreport](#), or [Rectangle](#) contains a data region or subreport that contains a **RepeatWith** property.

## 6.100 rsInvalidReportDefinition

*rsInvalidReportDefinition* is a critical error that occurs if a **report definition** is invalid. For example, if a [Report](#) contains an empty [Body](#) element, this error occurs.

## 6.101 rsInvalidReportParameterDependency

*rsInvalidReportParameterDependency* is a critical error that will occur when a [ReportParameter.DefaultValue](#) element or a [ValidValues.ParameterValues](#) element within a [ReportParameter](#) element has a forward dependency on another **ReportParameter** element.

## 6.102 rsInvalidRowGrouping

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsInvalidRowGrouping* is a critical error that will occur if a [RowGrouping](#) element contains both or neither of the following: [RowGrouping.StaticRows](#) and [RowGrouping.DynamicRows](#).

## 6.103 rsInvalidRunningValueAggregate

*rsInvalidRunningValueAggregate* is a critical error that will occur if a [RunningValue](#) aggregate function has as its *Function* argument either "Aggregate", "Previous", or "CountRows".

## 6.104 rsInvalidScopeInMatrix

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsInvalidScopeInMatrix* is a critical error that will occur when a [RunningValue](#), [RowNumber](#), or [Previous](#) aggregate function within a [Matrix](#) does not contain as its *Scope* parameter a string constant that is equal to the name of a containing group in the **Matrix**.

## 6.105 rsInvalidScopeInTablix

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsInvalidScopeInTablix* is a critical error that will occur when a [RunningValue](#), [RowNumber](#), or [Previous](#) aggregate function within a [Tablix](#) does not contain as its *Scope* parameter a string constant that is equal to the name of a containing group in the **Tablix**.

## 6.106 rsInvalidSeriesGrouping

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsInvalidSeriesGrouping* is a critical error that will occur when a [SeriesGrouping](#) element contains neither or both of the following elements: [SeriesGrouping.StaticSeries](#) and [SeriesGrouping.DynamicSeries](#).

## 6.107 rsInvalidStaticDataGrouping

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsInvalidStaticDataGrouping* is a critical error that will occur when a [DataGrouping.Grouping](#) element contains a [Grouping.GroupExpressions](#) element and [DataGrouping.Static](#) is set to true.

## 6.108 rsInvalidReportName

*rsInvalidReportName* is a critical error that will occur when the value of [Subreport.ReportName](#) or [Drillthrough.ReportName](#) is an empty string.

## 6.109 rsInvalidReportNameCharacters

*rsInvalidReportNameCharacters* is a critical error that will occur when the name of a [Report](#) contains one or more of the following invalid characters:

- ;
- ?
- :
- @
- &
- =
- +
- \$
- ,
- \
- \*
- <
- >

- |
- "

### 6.110 rsInvalidReportUri

*rsInvalidReportUri* is a critical error that will occur when the name of a [Report](#) contains characters that are invalid for a URI.

### 6.111 rsInvalidTargetScope

*rsInvalidTargetScope* is a critical error that occurs if a [UserSort.SortTarget](#) is set to a value that does not equal the name of the current scope, a peer scope that is a data region, or an ancestor scope.

### 6.112 rsInvalidTextboxInPageSection

*rsInvalidTextboxInPageSection* is a critical error that will occur when a [Textbox.UserSort](#) element exists within a [PageSection](#) or a [PageHeaderFooter](#) element.

### 6.113 rsInvalidReportItemInPageSection

**Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)**

*rsInvalidReportItemInPageSection* is a non-critical error that will occur when a [Tablix.DocumentMapLabel](#) element, [Chart.DocumentMapLabel](#) element, [GaugePanel.DocumentMapLabel](#) element or [Subreport.DocumentMapLabel](#) element exists within a [PageSection](#) element or a [PageHeaderFooter](#) element.

### 6.114 rsInvalidToggleItem

*rsInvalidToggleItem* is a critical error that occurs whenever a report item's [Visibility.ToggleItem](#) element is not a [Textbox](#) that is in the same scope or in a scope containing the report item, or if the report item's **grouping scope** does not contain a [Group.Parent](#) element.

### 6.115 rsInvalidValidValues

*rsInvalidValidValues* is a critical error that will occur whenever a [ValidValues](#) element contains either none or both of the following elements:

- [ValidValues.ParameterValues](#)
- [ValidValues.DataSetReference](#)

### 6.116 rsInvalidMultiValueParameter

*rsInvalidMultiValueParameter* is a critical error that will occur whenever a [ReportParameter](#) has both [ReportParameter.MultiValue](#) and [ReportParameter.Nullable](#) set to true.

### 6.117 rsMapLayerMissingProperty

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

*rsMapLayerMissingProperty* is a critical error that will occur when a required element in [MapLineLayer](#), [MapPointLayer](#), or [MapPolygonLayer](#) is missing.

## 6.118 rsMapPropertyAlreadyDefined

*Applies to [RDL 2010/01](#) and [RDL 2016/01](#)*

*rsMapPropertyAlreadyDefined* is a critical error that will occur when a [Map](#) subelement is already defined.

## 6.119 rsMissingAggregateScope

*rsMissingAggregateScope* is a critical error that occurs when an expression uses an aggregate function without a scope, that expression is used outside a data region, and the [Report](#) does not contain exactly one [DataSet](#).

## 6.120 rsMissingChartDataPoints

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsMissingChartDataPoints* is a critical error that will occur when a [Chart](#) does not contain a [ChartData](#) element.

## 6.121 rsMissingDataSetName

*rsMissingDataSetName* is a critical error that will occur when a data region is missing its **DataSetName** element, all of its ancestor data regions are missing their **DataSetName** elements, and there is more than one [DataSet](#) element within the containing [Report](#). For example, if a [List](#) is missing the [List.DataSetName](#) child element, is not the descendent of any other data regions, and the containing Report has multiple DataSet elements, this error will occur.

## 6.122 rsMissingMIMETYPE

*rsMissingMIMETYPE* is a critical error that will occur when an element that requires a [ReportMIMETYPE](#) value is not present within its parent element. For example, if an [Image](#) is missing its [Image.MIMETYPE](#) child element and the value of the peer [Image.Source](#) element is set to "Database", this error will occur.

## 6.123 rsMissingParameterDefault

*rsMissingParameterDefault* is a critical error that will occur when a [ReportParameter.DefaultValue](#) element is not present for a [ReportParameter](#) element that is either missing a [ReportParameter.Prompt](#) child element and has [ReportParameter.Nullable](#) set to false, or none of its [ValidValues.ParameterValues](#) elements within [ValidValues](#) is set to NULL.

## 6.124 rsMultipleGroupExpressionsOnChartScalarAxis

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsMultipleGroupExpressionsOnChartScalarAxis* is a critical error that will occur if an [Axis.Scalar](#) element is set to true for a [CategoryAxis](#) and the **DynamicCategory** associated with that **CategoryAxis** contains more than one [GroupExpressions.GroupExpression](#) element.

## 6.125 rsMultipleGroupingsOnChartScalarAxis

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsMultipleGroupingsOnChartScalarAxis* is a critical error that will occur if an [Axis.Scalar](#) element is set to true for a [CategoryAxis](#) and more than one level of [Grouping](#) elements are associated with that [CategoryAxis](#).

### 6.126 rsMultiReportItemsInMatrixSection

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

*rsMultiReportItemsInMatrixSection* is a critical error that will occur if a [DynamicColumns.ReportItems](#), [DynamicRows.ReportItems](#), or [MatrixCell.ReportItems](#) contains more than exactly one child element.

### 6.127 rsMultiReportItemsInTablixCell

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

*rsMultiReportItemsInTablixCell* is a critical error that will occur if a [TableCell.ReportItems](#) element contains more than exactly one child element.

### 6.128 rsMultiReportItemsInPageSectionExpression

*rsMultiReportItemsInPageSectionExpression* is a critical error that occurs when an expression within a [PageSection](#) or [PageHeaderFooter](#) element contains more than one reference to the [ReportItems](#) **global collection**.

### 6.129 rsMultiReportItemsInCustomReportItem

*rsMultiReportItemsInCustomReportItem* is a critical error that will occur when a [CustomReportItem.AltReportItem](#) contains more than one child element.

### 6.130 rsMultiStaticCategoriesOrSeries

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

*rsMultiStaticCategoriesOrSeries* is a critical error that will occur if a [Chart](#) contains multiple instances of either [SeriesGrouping.StaticSeries](#) elements or [CategoryGrouping.StaticCategories](#) elements.

### 6.131 rsMultiStaticColumnsOrRows

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

*rsMultiStaticColumnsOrRows* is a critical error that will occur if a [Matrix](#) contains multiple instances of either [RowGrouping.StaticRows](#) elements or [ColumnGrouping.StaticColumns](#) elements.

### 6.132 rsNegativeLeftWidth

*rsNegativeLeftWidth* is a critical error that will occur when the [Line.Left](#) and [Line.Width](#) elements of a [Line](#) add up to a negative number.

### 6.133 rsNegativeTopHeight

*rsNegativeTopHeight* is a critical error that will occur when the [Line.Top](#) and [Line.Height](#) elements of a [Line](#) add up to a negative number.

### 6.134 rsNonAggregateInMatrixCell

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

*rsNonAggregateInMatrixCell* is a non-critical error that will occur when a [MatrixCell](#) element contains an expression that contains a field that is not an aggregate function.

### 6.135 rsNonExistingScope

*rsNonExistingScope* is a critical error that will occur when a [UserSort.SortExpressionScope](#) element or [UserSort.SortTarget](#) contains a value that is not equal to the name of any [Group](#) or [DataSet](#) within a [Report](#).

### 6.136 rsNotAReportDefinition

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

*rsNotAReportDefinition* is a critical error that will occur if a [Report](#) definition is not provided. For example, if a **Report** element is missing from the .rdl file, this error will occur.

### 6.137 rsNotACurrentReportDefinition

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

*rsNotACurrentReportDefinition* is a critical error that will occur if the namespace of a [Report](#) does not match the current **Report** definition namespace.

### 6.138 rsOverlappingReportItems

*rsOverlappingReportItems* is a non-critical error that occurs when, within a [Report](#), two or more report item elements overlap in their area consumed. For example, if two [Textbox](#) elements overlap within a **Report**, this error occurs.

### 6.139 rsReportItemOutsideContainer

*rsReportItemOutsideContainer* is a non-critical error that occurs when a contained report item has its area extend outside the bounds of its container report item. For example, this non-critical error will occur whenever a [Textbox](#) extends past the edge of its containing [Rectangle](#).

### 6.140 rsPageBreakOnMatrixColumnGroup

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

*rsPageBreakOnMatrixColumnGroup* is a non-critical error that will occur if, for a [Matrix](#), any [Grouping](#) within the [DynamicColumns](#) element has the [Grouping.PageBreakAtStart](#) or [Grouping.PageBreakAtEnd](#) element set to true.

### 6.141 rsPageBreakOnChartGroup

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

*rsPageBreakOnChartGroup* is a non-critical error that will occur if, for a [Chart](#), any [Grouping](#) has the [Grouping.PageBreakAtStart](#) or [Grouping.PageBreakAtEnd](#) element set to true, or the [PageBreak.BreakLocation](#) element within the **Grouping** is not set to "None" or omitted.

## 6.142 rsParameterValueNullOrBlank

*rsParameterValueNullOrBlank* is a critical error that will occur if [ReportParameter.Nullable](#) or [ReportParameter.AllowBlank](#) is set to false, but a property within the [ReportParameter](#) violates that setting. For example, if the **ReportParameter.AllowBlank** is set to false, but a [Values.Value](#) within [ReportParameter.DefaultValue](#) is empty, this error will occur.

## 6.143 rsPreviousAggregateInFilterExpression

*rsPreviousAggregateInFilterExpression* is a critical error that will occur when a [Filter.FilterExpression](#) or [Filter.FilterValues](#) contains a reference to the [Previous](#) aggregate function.

## 6.144 rsPreviousAggregateInGroupExpression

*rsPreviousAggregateInGroupExpression* is a critical error that will occur when a [Group.GroupExpressions](#) contains a reference to the [Previous](#) aggregate function.

## 6.145 rsPreviousAggregateInPageSectionExpression

*rsPreviousAggregateInPageSectionExpression* is a critical error that will occur when an expression within a [PageSection](#) or [PageHeaderFooter](#) contains a reference to the [Previous](#) aggregate function.

## 6.146 rsPreviousAggregateInQueryParameterExpression

*rsPreviousAggregateInQueryParameterExpression* is a critical error that will occur when a [QueryParameter.Value](#) element contains a reference to the [Previous](#) aggregate function.

## 6.147 rsPreviousAggregateInReportParameterExpression

*rsPreviousAggregateInReportParameterExpression* is a critical error that will occur when an expression within a [ReportParameter](#) element contains a reference to the [Previous](#) aggregate function.

## 6.148 rsPreviousAggregateInReportLanguageExpression

*rsPreviousAggregateInReportLanguageExpression* is a critical error that will occur when an expression within a [Report.Language](#) element contains a reference to the [Previous](#) aggregate function.

## 6.149 rsPreviousAggregateInSortExpression

*rsPreviousAggregateInSortExpression* is a critical error that will occur whenever a [UserSort.SortExpression](#) or [SortExpression.Value](#) element contains a reference to the [Previous](#) aggregate function.

## 6.150 rsRepeatWithNotPeerDataRegion

*rsRepeatWithNotPeerDataRegion* is a critical error that occurs when a report item has the **RepeatWith** element specified and does not refer to the name of a peer data region.

## 6.151 rsReportItemInFilterExpression

*rsReportItemInFilterExpression* is a critical error that will occur when a [Filter.FilterExpression](#) element or [Filter.FilterValues](#) references the [ReportItems](#) global collection.

### 6.152 rsReportItemInGroupExpression

*rsReportItemInGroupExpression* is a critical error that will occur when a [Group.GroupExpressions](#) element references the [ReportItems](#) global collection.

### 6.153 rsReportItemInQueryParameterExpression

*rsReportItemInQueryParameterExpression* is a critical error that will occur when an expression within a [QueryParameter.Value](#) element refers to a [ReportItems](#) global collection.

### 6.154 rsReportItemInReportParameterExpression

*rsReportItemInReportParameterExpression* is a critical error that will occur when an expression within a [ReportParameter](#) element refers to a [ReportItems](#) global collection.

### 6.155 rsReportItemInSortExpression

*rsReportItemInSortExpression* is a critical error that will occur when a [UserSort.SortExpression](#) or a [SortExpression.Value](#) element refers to a [ReportItems](#) global collection.

### 6.156 rsReportItemInReportLanguageExpression

*rsReportItemInReportLanguageExpression* is a critical error that will occur when an expression within a [Report.Language](#) element refers to a [ReportItems](#) global collection.

### 6.157 rsReportItemInVariableExpression

*rsReportItemInVariableExpression* is a critical error that occurs when an expression within a [Variable.Value](#) element refers to a report item via the [ReportItems](#) global collection.

### 6.158 rsReportParameterPropertyTypeMismatch

*Applies to [RDL 2003/10](#), [RDL 2005/01](#), and [RDL 2008/01](#)*

*rsReportParameterPropertyTypeMismatch* is a critical error that will occur when a [ReportParameter.DataType](#) element does not match a [Values.Value](#) element that is set within [ReportParameter.DefaultValue](#) or if the **ReportParameter.DataType** element does not match a [ParameterValue.Value](#) element within [ReportParameter.ValidValues](#).

### 6.159 rsRowNumberInFilterExpression

*rsRowNumberInFilterExpression* is a critical error that will occur when a [Filter.FilterExpression](#) or [Filter.FilterValues](#) element uses the [RowNumber](#) aggregate function.

### 6.160 rsRowNumberInPageSectionExpression

*rsRowNumberInPageSectionExpression* is a critical error that will occur when an expression within a [PageSection](#) or [PageHeaderFooter](#) element uses the [RowNumber](#) aggregate function.

### 6.161 rsRowNumberInQueryParameterExpression

*rsRowNumberInQueryParameterExpression* is a critical error that will occur when a [QueryParameter.Value](#) element uses the [RowNumber](#) aggregate function.



### **6.162 rsRowNumberInReportParameterExpression**

*rsRowNumberInReportParameterExpression* is a critical error that will occur when an expression within a [ReportParameter](#) element uses the [RowNumber](#) aggregate function.

### **6.163 rsRowNumberInReportLanguageExpression**

*rsRowNumberInReportLanguageExpression* is a critical error that will occur when an expression within a [Report.Language](#) element uses the [RowNumber](#) aggregate function.

### **6.164 rsRowNumberInSortExpression**

*rsRowNumberInSortExpression* is a critical error that will occur when a [UserSort.SortExpression](#) or a [SortExpression.Value](#) element uses the [RowNumber](#) aggregate function.

### **6.165 rsRowNumberInVariableExpression**

*rsRowNumberInVariableExpression* is a critical error that will occur when a [Variable.Value](#) element uses the [RowNumber](#) aggregate function.

### **6.166 rsRunningValueInFilterExpression**

*rsRunningValueInFilterExpression* is a critical error that will occur when a [Filter.FilterExpression](#) element uses the [RunningValue](#) aggregate function.

### **6.167 rsRunningValueInGroupExpression**

*rsRunningValueInGroupExpression* is a critical error that will occur when a [Group.GroupExpressions](#) element uses the [RunningValue](#) aggregate function.

### **6.168 rsRunningValueInPageSectionExpression**

*rsRunningValueInPageSectionExpression* is a critical error that will occur when an expression within a [PageSection](#) or [PageHeaderFooter](#) element uses the [RunningValue](#) aggregate function.

### **6.169 rsRunningValueInQueryParameterExpression**

*rsRunningValueInQueryParameterExpression* is a critical error that will occur when a [QueryParameter.Value](#) element uses the [RunningValue](#) aggregate function.

### **6.170 rsRunningValueInReportParameterExpression**

*rsRunningValueInReportParameterExpression* is a critical error that will occur when an expression within a [ReportParameter](#) element uses the [RunningValue](#) aggregate function.

### **6.171 rsRunningValueInReportLanguageExpression**

*rsRunningValueInReportLanguageExpression* is a critical error that will occur when an expression within a [Report.Language](#) element uses the [RunningValue](#) aggregate function.

### **6.172 rsRunningValueInSortExpression**

*rsRunningValueInSortExpression* is a critical error that will occur when a [UserSort.SortExpression](#) or a [SortExpression.Value](#) element uses the [RunningValue](#) aggregate function.

### 6.173 rsRunningValueInVariableExpression

*rsRunningValueInVariableExpression* is a critical error that will occur when a [Variable.Value](#) element uses the [RunningValue](#) aggregate function.

### 6.174 rsScopeInPageSectionExpression

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsScopeInPageSectionExpression* is a critical error that will occur when an expression within a [PageHeaderFooter](#) element contains an aggregate function with a provided *Scope* parameter.

### 6.175 rsStaticGroupingOnChartScalarAxis

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsStaticGroupingOnChartScalarAxis* is a critical error that will occur if an [Axis.Scalar](#) element is set to true for a [CategoryAxis](#) and the **CategoryAxis** is associated only with a static [Grouping](#).

### 6.176 rsToggleInPageSection

*rsToggleInPageSection* is a critical error that will occur when a [Visibility.ToggleItem](#) element occurs within a [PageSection](#) or [PageHeaderFooter](#) element.

### 6.177 rsUnsortedCategoryInAreaChart

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsUnsortedCategoryInAreaChart* is a critical error that will occur if, for a [Chart](#) with [Chart.Type](#) set to "Area" and [Axis.Scalar](#) set to true within [Chart.CategoryAxis](#), the [SortBy.SortExpression](#) is not identical to the value of [Grouping.GroupExpressions](#) within the same [DynamicCategories](#) element.

### 6.178 rsWrongNumberOfMatrixCells

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsWrongNumberOfMatrixCells* is a critical error that will occur when a [MatrixRow](#) element contains a different number of [MatrixCell](#) elements than the number of [StaticColumn](#) elements within a [Matrix](#). If the matrix contains no **StaticColumn** instances, each **MatrixRow** will contain exactly one **MatrixCell** element.

### 6.179 rsWrongNumberOfMatrixColumns

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsWrongNumberOfMatrixColumns* is a critical error that will occur when a [Matrix](#) contains a different number of [MatrixColumn](#) elements than the number of [StaticColumn](#) elements. If the matrix contains no **StaticColumn** elements, the matrix will contain exactly one **MatrixColumn** element.

### 6.180 rsWrongNumberOfMatrixRows

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsWrongNumberOfMatrixRows* is a critical error that will occur when a [Matrix](#) contains a different number of [MatrixRow](#) elements than the number of [StaticRow](#) elements. If the matrix contains no **StaticRow** element, the matrix will contain exactly one **MatrixRow** element.

## 6.181 rsWrongNumberOfChartDataPoints

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsWrongNumberOfChartDataPoints* is a critical error that will occur when a [Chart.ChartData](#) element contains a different number of [DataPoint](#) elements than the product of the maximum of the number of [SeriesGrouping.StaticSeries](#) elements and 1, and the maximum of the number of [CategoryGrouping.StaticCategories](#) and 1.

## 6.182 rsWrongNumberOfChartSeries

*rsWrongNumberOfChartSeries* is a critical error that will occur when the number of [ChartSeries](#) elements within a [ChartSeriesCollection](#) element does not match the number of leaf-node [ChartMember](#) elements within a [Chart.ChartSeriesHierarchy](#), or when the number of [ChartData.ChartSeries](#) does not match the number of [SeriesGrouping.StaticSeries](#) for a [Chart](#). (A leaf-node **ChartMember** does not contain a child [ChartMember.ChartMembers](#) element.)

## 6.183 rsWrongNumberOfChartDataPointsInSeries

*rsWrongNumberOfChartDataPointsInSeries* is a critical error that will occur if a [Chart](#) contains a different number of [DataPoint](#) elements per [ChartSeries](#) element than the number of leaf [StaticCategories.StaticMember](#) elements within the **Chart**.

## 6.184 rsWrongNumberOfDataValues

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsWrongNumberOfDataValues* is a critical error that will occur if the number of [DataValue](#) elements within a [DataPoint.DataValues](#) for a [Chart](#) is incompatible for the [Chart.Type](#). For example, if **Chart.Type** is set to "Bubble" but a **DataPoint.DataValues** collection within that **Chart** does not contain exactly three **DataValue** elements, this error will occur.

## 6.185 rsWrongNumberOfParameters

*rsWrongNumberOfParameters* is a critical error that will occur when an expression uses one of the aggregate functions with an incorrect number of parameters. For example, this error will occur when a [TextRun.Value](#) element contains an expression "=Sum()".

## 6.186 rsWrongNumberOfTableCells

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsWrongNumberOfTableCells* is a critical error that will occur when a [TableRow](#) element contains a different number of [TableCell](#) elements than the number of [TableColumn](#) elements for the same [Table](#).

## 6.187 rsMissingDataGrouping

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsMissingDataGrouping* is a critical error that will occur when a [CustomReportItem](#) contains [DataRow.DataCell](#) elements but does not contain corresponding [DataRowGroupings.DataGroupings](#) elements.

## 6.188 rsWrongNumberOfDataRows

*rsWrongNumberOfDataRows* is a critical error that will occur if a [CustomReportItem.CustomData](#) element contains a different number of [DataRow](#) elements within its [DataRows](#) collection than the number of leaf-node [DataGrouping](#) elements within the [DataRowGroupings](#) collection. (A leaf-node **DataGrouping** element does not contain a child [DataGrouping.DataGroupings](#) element.)

## 6.189 rwWrongNumberOfDataCellsInDataRow

*rwWrongNumberOfDataCellsInDataRow* is a critical error that will occur if a [CustomReportItem.CustomData](#) element contains a different number of [DataRow.DataCell](#) elements within a [DataRows](#) collection than the number of leaf-node [DataGrouping](#) elements within the [DataColumnGroupings](#) collection. (A leaf-node **DataGrouping** element does not contain a child [DataGrouping.DataGroupings](#) element.)

## 6.190 rsInvalidRecursiveAggregate

*rsInvalidRecursiveAggregate* is a critical error that will occur when an expression contains either a [First](#), [Last](#), [Previous](#), [RowNumber](#), [RunningValue](#), or [Aggregate](#) function with the *Recursive* function parameter present.

## 6.191 rsInvalidAggregateRecursiveFlag

*rsInvalidAggregateRecursiveFlag* is a critical error that will occur when one of the aggregate functions contains as its *Recursive* parameter a value other than "Recursive" or "Simple".

## 6.192 rsPostSortAggregateInGroupFilterExpression

*rsPostSortAggregateInGroupFilterExpression* is a critical error that will occur when a [First](#), [Last](#), or [Previous](#) aggregate function is present within a [Filter.FilterExpression](#) or a [Filter.FilterValues](#) element within a [Group](#) element.

## 6.193 rsPostSortAggregateInSortExpression

*rsPostSortAggregateInSortExpression* is a critical error that will occur when a [First](#), [Last](#), [Previous](#) aggregate function is present within a [UserSort.SortExpression](#) or a [SortExpression.Value](#) element within a [TablixMember](#), [ChartMember](#), [GaugeMember](#), [DataMember](#), or [CustomData](#) element.

## 6.194 rsPostSortAggregateInVariableExpression

*rsPostSortAggregateInVariableExpression* is a critical error that will occur when a [First](#), [Last](#), or [Previous](#) aggregate function is present within a [Variable.Value](#) element.

## 6.195 rsAggregateInPreviousAggregate

*rsAggregateInPreviousAggregate* is a critical error that will occur when a [Previous](#) aggregate function uses within its *Expression* parameter an [Aggregate](#) aggregate function.

## 6.196 rsRunningValueInPreviousAggregate

*rsRunningValueInPreviousAggregate* is a critical error that will occur when a [Previous](#) aggregate function uses within its *Expression* parameter a [RunningValue](#) aggregate function.

### 6.197 rsPreviousInPreviousAggregate

*rsPreviousInPreviousAggregate* is a critical error that will occur when a [Previous](#) aggregate function uses within its *Expression* parameter a **Previous** aggregate function.

### 6.198 rsRowNumberInPreviousAggregate

*rsRowNumberInPreviousAggregate* is a critical error that will occur when a [Previous](#) aggregate function uses within its *Expression* parameter a [RowNumber](#) aggregate function.

### 6.199 rsInScopeOrLevelInPreviousAggregate

*rsInScopeOrLevelInPreviousAggregate* is a critical error that will occur when a [Previous](#) aggregate function uses within its *Expression* parameter an [InScope](#) or [Level](#) function.

### 6.200 rsInvalidScopeInInnerAggregateOfPreviousAggregate

*rsInvalidScopeInInnerAggregateOfPreviousAggregate* is a critical error that occurs when a [Previous](#) aggregate function uses within its *Expression* parameter an aggregate function whose scope is not equal to or is not contained by the scope that is specified by the *PreviousScope* parameter.

### 6.201 rsInvalidGroupingParent

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

*rsInvalidGroupingParent* is a critical error that will occur when a [Grouping](#) element contains a [Grouping.Parent](#) element but more than one [Grouping.GroupExpressions](#) elements.

### 6.202 rsMissingDataGroupings

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

*rsMissingDataGroupings* is a critical error that will occur when a [CustomReportItem](#) does not contain at least one [DataColumnGroupings.DataGroupings](#) element and one [DataRowGroupings.DataGroupings](#) element.

### 6.203 rsMissingDataCells

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

*rsMissingDataCells* is a critical error that will occur when a [CustomReportItem](#) does not contain at least one [DataRow](#) element with at least one [DataCell](#) element.

### 6.204 rsCRIMultiStaticColumnsOrRows

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

*rsCRIMultiStaticColumnsOrRows* is a critical error that will occur when a [CustomReportItem](#) contains at least two [DataGrouping](#) elements with [DataGrouping.Static](#) set to true along the same axis, such as a row or column axis, but at different levels.

### 6.205 rsCRISStaticWithSubgroups

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

*rsCRISubtotalNotSupported* is a critical error that will occur when a [CustomReportItem](#) contains a [DataGrouping](#) element that has [DataGrouping.Static](#) set to true and [DataGrouping.DataGroupings](#) specified.

## 6.206 rsCRIMultiNonStaticGroups

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsCRIMultiNonStaticGroups* is a critical error that will occur when a [CustomReportItem](#) with at least one [DataGrouping](#) element with [DataGrouping.Static](#) set to true and at least one **DataGrouping** element with **DataGrouping.Static** set to false occur within the same [DataGroupings](#) collection.

## 6.207 rsCRISubtotalNotSupported

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsCRISubtotalNotSupported* is a critical error that will occur when a [CustomReportItem](#) contains a [DataGrouping](#) with [DataGrouping.Subtotal](#) set to true.

## 6.208 rsInvalidGrouping

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsInvalidGrouping* is a critical error that will occur when a [Grouping](#) element is present within a [CustomReportItem](#) without a [Grouping.GroupExpressions](#).

## 6.209 rsCRIInPageSection

*rsCRIInPageSection* is a critical error that will occur when a [CustomReportItem](#) is present inside a [PageSection](#) or [PageHeaderFooter](#) element.

## 6.210 rsBookmarkInPageSection

*rsBookmarkInPageSection* is a non-critical error that occurs when a report item contains a **Bookmark** property within a [PageSection](#) or [PageHeaderFooter](#) element.

## 6.211 rsCantMakeTableGroupHeadersFixed

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsCantMakeTableGroupHeadersFixed* is a critical error that will occur if a [TableGroup.Header](#) has [Header.FixedHeader](#) set to true.

## 6.212 rsFixedHeadersInInnerDataRegion

*Applies to [RDL 2005/01](#), [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsFixedHeadersInInnerDataRegion* is a critical error that occurs if a [Tablix](#) has [Tablix.FixedColumnHeaders](#), [Tablix.FixedRowHeaders](#), or [TablixMember.FixedData](#) set to true but the **Tablix** exists within another data region. This critical error also occurs if a [Header.FixedHeader](#) is set to true for a [Table](#) that exists within another **Table**, [List](#), or [Matrix](#).

## 6.213 rsInvalidFixedTableColumnHeaderSpacing

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsInvalidFixedTableColumnHeaderSpacing* is a critical error that will occur if, for a [TableColumns](#) collection, the set of [TableColumn](#) elements with [TableColumn.FixedHeader](#) set to true are not contiguous or if the **TableColumns** collection does not include the first or last **TableColumn** in the collection.

#### **6.214 rsUnsupportedProtocol**

*rsUnsupportedProtocol* is a critical error that will occur when an [Image.Value](#) element that requires an [RdlURL](#) value does not begin with "https://", "http://", "ftp://", "mailto:", or "news:".

#### **6.215 rsVariableInPreviousAggregate**

*rsVariableInPreviousAggregate* is a critical error that will occur when a [Previous](#) aggregate function has within its *Expression* parameter a reference to a [Variables](#) global collection.

#### **6.216 rsAggregateOfVariable**

*rsAggregateOfVariable* is a critical error that will occur when one of the aggregate functions in an expression has within any of its function parameters a reference to a [Variables](#) global collection.

#### **6.217 rsVariableInQueryParameterExpression**

*rsVariableInQueryParameterExpression* is a critical error that will occur when a [QueryParameter.Value](#) element contains a reference to a [Variables](#) global collection.

#### **6.218 rsVariableInReportParameterExpression**

*rsVariableInReportParameterExpression* is a critical error that will occur when an expression within a [ReportParameter](#) element contains a reference to a [Variables](#) global collection.

#### **6.219 rsVariableInReportLanguageExpression**

*rsVariableInReportLanguageExpression* is a critical error that will occur when an expression within a [Report.Language](#) element contains a reference to a [Variables](#) global collection.

#### **6.220 rsVariableInGroupExpression**

*rsVariableInGroupExpression* is a critical error that will occur when a [Group.GroupExpressions](#) element contains a reference to a [Variables](#) global collection.

#### **6.221 rsVariableInCalculatedFieldExpression**

*rsVariableInCalculatedFieldExpression* is a critical error that will occur when a [Field.Value](#) element contains a reference to a [Variables](#) global collection.

#### **6.222 rsDataSetInPageSectionExpression**

**Applies to [RDL 2003/10](#) and [RDL 2005/01](#)**

*rsDataSetInPageSectionExpression* is a critical error that will occur when a [PageHeaderFooter](#) element contains an expression that references a [DataSets](#) global collection.



### 6.223 rsDataSetInQueryParameterExpression

*rsDataSetInQueryParameterExpression* is a critical error that will occur when a [QueryParameter.Value](#) element contains an expression that references a [DataSets](#) global collection.

### 6.224 rsDataSetInReportParameterExpression

*rsDataSetInReportParameterExpression* is a critical error that will occur when a [ReportParameter](#) element contains an expression that references a [DataSets](#) global collection.

### 6.225 rsDataSetInReportLanguageExpression

*rsDataSetInReportLanguageExpression* is a critical error that will occur when a [Report.Language](#) element contains an expression that references a [DataSets](#) global collection.

### 6.226 rsDataSourceInPageSectionExpression

*Applies to [RDL 2003/10](#) and [RDL 2005/01](#)*

*rsDataSetInReportLanguageExpression* is a critical error that will occur when an expression within a [PageHeaderFooter](#) element references a [DataSource](#) via the [DataSources](#) global collection.

### 6.227 rsDataSourceInQueryParameterExpression

*rsDataSourceInQueryParameterExpression* is a critical error that will occur when a [QueryParameter.Value](#) element references a [DataSources](#) global collection.

### 6.228 rsDataSourceInReportParameterExpression

*rsDataSourceInReportParameterExpression* is a critical error that will occur when an expression within a [ReportParameter](#) element references a [DataSource](#) via the [DataSources](#) global collection.

### 6.229 rsDataSourceInReportLanguageExpression

*rsDataSourceInReportLanguageExpression* is a critical error that will occur when an expression within a [Report.Language](#) element references a [DataSource](#) via the [DataSources](#) global collection.

### 6.230 rsInvalidMeDotValueInExpression

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsInvalidMeDotValueInExpression* is a critical error that will occur when an expression in a child element of a [Paragraph](#) contains the string "Me.Value".

### 6.231 rsWrongNumberOfTablixCornerRows

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsWrongNumberOfTablixCornerRows* is a critical error that will occur when the number of [TablixCornerRow](#) elements for a [Tablix](#) is different from the number of distinct header rows created by the [TablixMember.TablixHeader](#) elements in the [Tablix.TablixColumnHierarchy](#).

### 6.232 rsWrongNumberOfTablixCornerCells

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*



*rsWrongNumberOfTablixCornerCells* is a critical error that will occur when the number of [TablixCornerCell](#) elements within a [TablixCornerRow](#) is different than the number of distinct header columns created by the [TablixMember.TablixHeader](#) elements in the [Tablix.TablixRowHierarchy](#).

### 6.233 rsWrongNumberOfTablixColumns

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsWrongNumberOfTablixColumns* is a critical error that will occur when the number of [TablixColumn](#) elements within a [Tablix](#) does not equal the number of innermost [TablixMember](#) elements (such as a **TablixMember** element that has no child **TablixMember** elements) in the [Tablix.TablixColumnHierarchy](#).

### 6.234 rsWrongNumberOfTablixCells

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsWrongNumberOfTablixCells* is a critical error that will occur when the number of [TablixCell](#) instances within a [TablixRow](#) does not equal the number of innermost [TablixMember](#) elements (such as a **TablixMember** element that has no child **TablixMember** elements) in the [Tablix.TablixColumnHierarchy](#).

### 6.235 rsWrongNumberOfTablixRows

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsWrongNumberOfTablixRows* is a critical error that will occur when the number of [TablixRow](#) instances within a [Tablix](#) does not equal the number of innermost [TablixMember](#) elements (such as a **TablixMember** element that has no child **TablixMember** elements) in the [Tablix.TablixRowHierarchy](#).

### 6.236 rsInvalidTablixCornerCellSpan

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsInvalidTablixCornerCellSpan* is a critical error that will occur when a [TablixCornerCell.CellContents](#), [CellContents.RowSpan](#), or [CellContents.ColSpan](#) element is set to "0" when [CellContents](#) actually contains items or is set to nonzero when **CellContents** is empty.

### 6.237 rsInvalidTablixCornerRowSpans

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsInvalidTablixCornerRowSpans* is a critical error that will occur when the combined value of all [CellContents.RowSpan](#) elements aligned with a particular [TablixColumn](#) element does not equal the number of [TablixMember](#) elements within the [TablixColumnHierarchy](#) that contain a [TablixMember.TablixHeader](#) child element.

### 6.238 rsInvalidTablixCornerColumnSpans

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsInvalidTablixCornerColumnSpans* is a critical error that will occur when the combined value of all [CellContents.ColSpan](#) elements within a [TablixCornerRow](#) element does not equal the number of [TablixMember](#) elements within the [TablixRowHierarchy](#) that contain a [TablixMember.TablixHeader](#) child element.

## 6.239 rsInvalidSortNotAllowed

*rsInvalidSortNotAllowed* is a critical error that will occur when a [ChartMember](#), [TablixMember](#), or [CustomReportItem](#) contains a [SortExpressions](#) element that does not have an associated [Group](#) defined.

## 6.240 rsInvalidFixedHeaderOnOppositeHierarchy

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidFixedHeaderOnOppositeHierarchy* is a critical error that will occur when one of the following conditions is true:

- [TablixMember.FixedData](#) is set to true within a [TablixRowHierarchy](#), and [Tablix.FixedColumnHeaders](#) is set to true.
- **TablixMember.FixedData** is set to true within a [TablixColumnHierarchy](#), and [Tablix.FixedRowHeaders](#) is set to true.

## 6.241 rsInvalidFixedDataColumnPosition

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidFixedDataColumnPosition* is a critical error that will occur when the first [TablixMember](#) within a [TablixColumnHierarchy](#) has [TablixMember.FixedData](#) set to true and [Tablix.GroupsBeforeRowHeaders](#) is set to true.

## 6.242 rsInvalidFixedDataRowPosition

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidFixedDataRowPosition* is a critical error that will occur when [TablixMember.FixedData](#) is set to true for a non-first [TablixMember](#) within a [TablixRowHierarchy](#).

## 6.243 rsInvalidFixedDataNotContiguous

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidFixedDataNotContiguous* is a critical error that will occur when peer [TablixMember](#) elements with [TablixMember.FixedData](#) set to true are not contiguous.

## 6.244 rsInvalidFixedDataInHierarchy

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidFixedDataInHierarchy* is a critical error that will occur when a [TablixMember.FixedData](#) element is set to true for [TablixMember](#) elements that have an ancestor **TablixMember** element.

## 6.245 rsInvalidKeepWithGroup

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidKeepWithGroup* is a critical error that will occur if a [TablixMember.KeepWithGroup](#) is set to "Before" or "After", and any sibling [TablixMember](#) in between the original **TablixMember** and the target dynamic **TablixMember** (with [TablixMember.Group](#) specified) has a different **TablixMember.KeepWithGroup** value than that of the original **TablixMember**.

## 6.246 rsInvalidKeepWithGroupOnDynamicTablixMember

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidKeepWithGroupOnDynamicTablixMember* is a critical error that will occur when a [TablixMember](#) with [TablixMember.Group](#) specified has [TablixMember.KeepWithGroup](#) set to a value other than "None".

## 6.247 rsInvalidKeepWithGroupOnColumnTablixMember

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidKeepWithGroupOnColumnTablixMember* is a critical error that will occur when a [TablixMember](#) instance within the [Tablix.TablixColumnHierarchy](#) has [TablixMember.KeepWithGroup](#) set to a value other than "None".

## 6.248 rsInvalidRepeatOnNewPageOnColumnTablixMember

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidRepeatOnNewPageOnColumnTablixMember* is a critical error that will occur when a [TablixMember](#) instance within the [Tablix.TablixColumnHierarchy](#) has [TablixMember.RepeatOnNewPage](#) set to a value that is not false.

## 6.249 rsInvalidRepeatOnNewPage

*rsInvalidRepeatOnNewPage* is a critical error that will occur if the following conditions are true:

- [TablixMember.KeepWithGroup](#) is set to "Before" or "After".
- Any sibling [TablixMember](#) between the original **TablixMember** and the target dynamic **TablixMember** (with [TablixMember.Group](#) specified) has a different [TablixMember.RepeatOnNewPage](#) value than that of the original **TablixMember**.

## 6.250 rsInvalidTablixCellColSpans

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidTablixCellColSpans* is a critical error that will occur when the combined value of the [CellContents.ColSpan](#) elements within a [TablixRow.TablixCells](#) element does not equal the number of leaf-node [TablixMember](#) (with no child [TablixMembers](#) element) instances in the [TablixColumnHierarchy](#). (A leaf-node **TablixMember** has no child **TablixMembers** element.)

## 6.251 rsInvalidTablixCellColSpan

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidTablixCellColSpan* is a critical error that will occur when a [TablixCell](#) has its [CellContents.ColSpan](#) set such that the **TablixCell** spans columns under [TablixMember](#) elements within the [TablixColumnHierarchy](#) that are not static peer **TablixMember** elements, or does not exist under the same dynamic **TablixMember** in the **TablixColumnHierarchy**.

## 6.252 rsInvalidTablixCellRowSpan

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidTablixCellRowSpan* is a critical error that will occur when a [TablixCell.CellContents](#) element has [CellContents.RowSpan](#) set to a value other than 1.

### 6.253 rsCellContentsNotOmitted

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsCellContentsNotOmitted* is a critical error that will occur when a [CellContents](#) element is present but is already spanned by another **CellContents** element.

### 6.254 rsCellContentsRequired

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsCellContentsRequired* is a critical error that will occur when a [CellContents](#) element is not present but is required because it is not spanned by another **CellContents** element.

### 6.255 rsInvalidTablixCellCellSpan

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidTablixCellCellSpan* is a critical error that will occur when a [TablixCell.CellContents](#), [CellContents.RowSpan](#), or [CellContents.ColSpan](#) element is set to "0" when [CellContents](#) actually contains items or is set to nonzero when **CellContents** is empty.

### 6.256 rsInconsistentNumberOfCellsInRow

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInconsistentNumberOfCellsInRow* is a critical error that will occur when two or more [TablixRow](#) elements within the same [TablixRows](#) collection contain a different number of [TablixCell](#) elements. Or, this error will occur when two or more [TablixCornerRow](#) elements within the same [TablixCornerRows](#) collection contain a different number of [TablixCornerCell](#) elements.

### 6.257 rsInvalidTablixHeaderSize

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidTablixHeaderSize* is a critical error that occurs when the [TablixHeader.Size](#) for a tablix header is invalid. Each header column has to be equal in total height in the [TablixColumnHierarchy](#), and each header row has to be equal in width in the [TablixRowHierarchy](#).

### 6.258 rsInvalidTablixHeaders

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidTablixHeaders* is a critical error that will occur when, within a set of sibling [TablixMember](#) elements, at least one **TablixMember** exists that does not have a [TablixMember.TablixHeader](#) element present, and no sibling **TablixMember** elements or their descendant **TablixMember** elements contain a [TablixHeader](#).

### 6.259 rsInvalidInnerDataSetName

*rsInvalidInnerDataSetName* is a critical error that occurs when a data region has a different **DataSetName** property than a child data region. For example, this error will occur if a [Tablix](#) has a

[Tablix.DataSetName](#) element whose value is different from the value of a [Chart.DataSetName](#) element of a contained [Chart](#).

## 6.260 rsDuplicateVariableName

*rsDuplicateVariableName* is a critical error that will occur when two [Variable](#) elements within the same [Report](#) or [Group](#) have the same **Name** attribute.

## 6.261 rsInvalidVariableReference

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsInvalidVariableReference* is a critical error that will occur when an expression refers to a [Variable](#) via the [Variables](#) global collection that is not declared in the [Report](#), in the same [Group](#), or in a containing **Group**.

## 6.262 rsInvalidVariableNameNotCLSCompliant

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsInvalidVariableNameNotCLSCompliant* is a critical error that will occur when the value of the **Name** attribute of a [Variable](#) element within [Report.Variables](#) is not CLS-compliant [\[UTR15\]](#).

## 6.263 rsInvalidVariableNameLength

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsInvalidVariableNameLength* is a critical error that will occur when the value of the **Name** attribute of a [Variable](#) element within [Report.Variables](#) is either "0" or longer than 256 characters.

## 6.264 rsInvalidGroupingVariableNameNotCLSCompliant

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsInvalidGroupingVariableNameNotCLSCompliant* is a critical error that will occur when the value of the **Name** attribute of a [Variable](#) element within a [Group.Variables](#) element is not CLS-compliant [\[UTR15\]](#).

## 6.265 rsInvalidGroupingVariableNameLength

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsInvalidGroupingVariableNameLength* is a critical error that will occur when the value of the **Name** attribute of a [Variable](#) element within a [Group.Variables](#) element is either "0" or longer than 256 characters.

## 6.266 rsInvalidVariableCount

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsInvalidVariableCount* is a critical error that will occur if a [Group.Variables](#) or [Report.Variables](#) collection does not contain any [Variable](#) elements.

## 6.267 rsMissingExpression

*Applies to [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)*

*rsMissingExpression* is a critical error that will occur if a [Variable.Value](#) element is empty.

## 6.268 rsInvalidActionsCount

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidActionsCount* is a critical error that will occur when an [ActionInfo](#) element contains more than one [Action](#) element.

## 6.269 rsInvalidFixedDataBodyCellSpans

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidFixedDataBodyCellSpans* is a critical error that will occur when the following conditions are true:

- A [Tablix](#) contains a set of [TablixMember](#) elements with [TablixMember.FixedData](#) set to true within a [TablixColumnHierarchy](#) element.
- Some [TablixCell](#) elements are invalid because their [CellContents.ColSpan](#) fall outside the range of the fixed column **TablixMember** elements.

## 6.270 rsInvalidEmptyImageReference

*rsInvalidEmptyImageReference* is a critical error that will occur when an [Image](#) has [Image.Source](#) set to "External" but [Image.Value](#) is set to either nothing or empty white space.

## 6.271 rsFieldReference

*rsFieldReference* is a critical error that occurs when an expression refers to a field that does not exist with the current [DataSet](#) scope, or, if inside an aggregate, a specified **DataSet** scope. For example, if an expression inside a [Tablix](#) that has [Tablix.DataSetName](#) set to "myDataSet" has the value "=Fields!myValue.value", but "myValue" is not the name of a [Field](#) in the "myDataSet" **DataSet**, this error occurs.

## 6.272 rsInvalidBackgroundRepeat

*rsInvalidBackgroundRepeat* is a critical error that will occur when a [BackgroundImage.BackgroundRepeat](#) element is not set to one of its valid values.

## 6.273 rsInvalidBackgroundGradientType

*rsInvalidBackgroundGradientType* is a critical error that occurs when a [Style.BackgroundGradientType](#) element is set to a constant [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is not one of its valid values.

## 6.274 rsInvalidBorderStyle

*rsInvalidBorderStyle* is a critical error that occurs when a [Border.Style](#) element is set to a constant [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is not one of its valid values.

## 6.275 rsInvalidCalender

*rsInvalidCalender* is a critical error that occurs when a [Style.Calendar](#) element is set to a constant [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is not one of its valid values.

## 6.276 rsInvalidCalendarForLanguage

*rsInvalidCalendarForLanguage* is a critical error that will occur when a [Style.Calendar](#) is incompatible in the Microsoft .NET Framework with a [Style.Language](#) for the same [Style](#) element.

## 6.277 rsInvalidColor

*rsInvalidColor* is a critical error that will occur when an element requiring an [RdlColor](#) value is not set to a valid value.

## 6.278 rsInvalidDirection

*rsInvalidDirection* is a critical error that occurs when a [Style.Direction](#) element is set to a constant [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is not one of its valid values.

## 6.279 rsInvalidEmbeddedImageProperty

*rsInvalidEmbeddedImageProperty* is a critical error that will occur when an [Image](#) or [Style.BackgroundImage](#) element specifies that the source of the image is embedded in the RDL but the name that is specified is not set to the name of an image in the [Report.EmbeddedImages](#) element.

## 6.280 rsInvalidFontStyle

*rsInvalidFontStyle* is a critical error that occurs when a [Style.FontStyle](#) element is set to a constant [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is not one of its valid values.

## 6.281 rsInvalidFontWeight

*rsInvalidFontWeight* is a critical error that occurs when a [Style.FontWeight](#) element is set to a constant [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is not one of its valid values.

## 6.282 rsInvalidLanguage

*rsInvalidLanguage* is a critical error that will occur when an element that requires a [ReportLanguage](#) value is not set to a valid value.

## 6.283 rsInvalidMeasurementUnit

**Applies to** [RDL 2003/10](#), [RDL 2005/01](#), and [RDL 2008/01](#)

*rsInvalidMeasurementUnit* is a critical error that will occur when an element that requires an [RdlSize](#) value is set to a measurement in terms of "em", "ex", or "%".

## 6.284 rsInvalidMIMETYPE

*rsInvalidMIMETYPE* is a critical error that occurs when an element that requires a [ReportMIMETYPE](#) value is not set to a valid value. Specifically, this error will occur if [EmbeddedImage.MIMETYPE](#) or [Image.MIMETYPE](#) for an [Image](#) that has [Image.Source](#) set to "Database" is not set to a valid [ReportMIMETYPE](#) value.

## 6.285 rsInvalidNumeralVariant

*rsInvalidNumeralVariant* is a critical error that occurs when a [Style.NumeralVariant](#) element is set to a constant [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is not one of its valid values.



## 6.286 rsInvalidNumeralVariantForLanguage

*rsInvalidNumeralVariantForLanguage* is a critical error that will occur when a [Style.NumeralVariant](#) element is set to a value that is incompatible with the value of the peer element [Style.Language](#).

## 6.287 rsInvalidSize

*rsInvalidSize* is a critical error that will occur if an element expecting an [RdlSize](#) value contains an expression that does not evaluate to a valid **RdlSize** value.

## 6.288 rsInvalidTextAlign

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

*rsInvalidTextAlign* is a critical error that occurs when a [Style.TextAlign](#) element is set to a constant [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is not one of its valid values.

## 6.289 rsInvalidTextDecoration

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

*rsInvalidTextDecoration* is a critical error that occurs when a [Style.TextDecoration](#) element is set to a constant [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is not one of its valid values.

## 6.290 rsInvalidUnicodeBiDi

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

*rsInvalidUnicodeBiDi* is a critical error that occurs when a [Style.UnicodeBiDi](#) element is set to a constant [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is not one of its valid values.

## 6.291 rsInvalidVerticalAlign

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

*rsInvalidVerticalAlign* is a critical error that occurs when a [Style.VerticalAlign](#) element is set to a constant [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is not one of its valid values.

## 6.292 rsInvalidWritingMode

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

*rsInvalidWritingMode* is a critical error that occurs when a [Style.WritingMode](#) element is set to a constant [String](#) ([\[XMLSCHEMA2/2\]](#) section 3.2.1) that is not one of its valid values.

## 6.293 rsNegativeSize

*rsNegativeSize* is a critical error that will occur when an element that requires a nonnegative [RdlSize](#) value contains a negative **RdlSize** value.

## 6.294 rsOutOfRangeSize

*rsOutOfRangeSize* is a critical error that will occur when an element that requires an [RdlSize](#) value contains an **RdlSize** value that is outside of its valid range. For example, this error will occur if [Style.PaddingTop](#) is set to "1001pt", even though **Style.PaddingTop** will not exceed "1001pt" or any equivalent **RdlSize** values.



## 6.295 rsPageNumberInBody

*rsPageNumberInBody* is a critical error that will occur when an expression contains the "PageNumber" or "TotalPages" variable from the Globals global collection, and that expression is not within a [PageSection](#) or [PageHeaderFooter](#) element.

## 6.296 rsParameterReference

*rsParameterReference* is a critical error that will occur when an expression refers to a nonexistent [ReportParameter](#) via the [Parameters](#) global collection.

## 6.297 rsReportItemReference

*rsReportItemReference* is a critical error that occurs when an expression refers to a nonexistent report item via the [ReportItems](#) global collection.

## 6.298 rsDataSetReference

*rsDataSetReference* is a critical error that will occur when an expression refers to a nonexistent [DataSet](#) via the [DataSets](#) global collection.

## 6.299 rsDataSourceReference

*rsDataSourceReference* is a critical error that will occur when an expression refers to a nonexistent [DataSource](#) via the [DataSources](#) global collection.

## 6.300 rsErrorLoadingCodeModule

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsErrorLoadingCodeModule* is a critical error that will occur if a [CodeModules.CodeModule](#) listed in the [Report.CodeModules](#) element fails to load.

## 6.301 rsInvalidObjectNameNotUnique

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidObjectNameNotUnique* is a critical error that will occur when an element within a collection in a [GaugePanel](#) or [Chart](#) element does not have a **Name** attribute with a unique or non-null value. For example, this error will occur if there exist two [RadialPointer](#) elements with their **Name** attribute set to "myRadialPointer" within the same [RadialPointers](#) collection.

## 6.302 rsInvalidObjectNameNotCLSCompliant

*rsInvalidObjectNameNotCLSCompliant* is a critical error that will occur when an element a [GaugePanel](#) or [Chart](#) has a **Name** attribute with a non CLS-compliant value [\[UTR15\]](#).

## 6.303 rsInvalidSourceSeriesName

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidSourceSeriesName* is a critical error that will occur when a [ChartDerivedSeries.SourceChartSeriesName](#) does not refer to the name of an existing [ChartSeries](#) within the [ChartData.ChartSeriesCollection](#) for a [Chart](#).

### 6.304 rsInvalidDataSourceNameNotCLSCompliant

*rsInvalidDataSourceNameNotCLSCompliant* is a critical error that will occur when the value of the **Name** attribute of a [DataSource](#) is not a **CLS-compliant identifier** [UTR15].

### 6.305 rsDuplicateChartLegendItemName

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsDuplicateChartLegendItemName* is a critical error that will occur if two or more [ChartLegendCustomItem](#) elements within the same [ChartLegendCustomItems](#) collection have the same **Name** attribute.

### 6.306 rsInvalidEnumValue

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidEnumValue* is a critical error that will occur when any element within a [GaugePanel](#) that requires a string value or expression from a list of valid values contains an invalid value. For example, [RadialPointer.NeedleStyle](#) might be set to "invalidStyle". However, "invalidStyle" is not a valid value for **RadialPointer.NeedleStyle**. Therefore, this error will occur.

### 6.307 rsInvalidListStyle

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidListStyle* is a critical error that will occur when a [Paragraph.ListStyle](#) element is not set to one of its valid values.

### 6.308 rsInvalidMarkupType

*Applies to* [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsInvalidMarkupType* is a critical error that will occur when a [TextRun.MarkupType](#) element is not set to one of its valid values.

### 6.309 rsMissingAggregateScopeInPageSection

*rsMissingAggregateScopeInPageSection* is a critical error that will occur when an expression within a [PageSection](#) or [PageHeaderFooter](#) contains an aggregate function that does not have its *Scope* parameter specified.

### 6.310 rsReportItemInScopedAggregate

*rsReportItemInScopedAggregate* is a critical error that occurs when the following conditions are true:

- An expression within a [PageSection](#) or [PageHeaderFooter](#) contains an aggregate function that has its *Scope* parameter set.
- That expression references a report item via the [ReportItems](#) global collection.

### 6.311 rsPageNumberInScopedAggregates

*rsPageNumberInScopedAggregates* is a critical error that will occur when the following conditions are true:

- An expression within a [PageSection](#) or [PageHeaderFooter](#) contains an aggregate function that has its *Scope* parameter set.
- That expression contains the "PageNumber" or "TotalPages" variable from the [Globals](#) global collection.

### 6.312 rsVariableInDataRowSortExpression

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsVariableInDataRowSortExpression* is a critical error that will occur if an expression within [GaugePanel.SortExpressions](#), [Chart.SortExpressions](#), or [Tablix.SortExpressions](#) references a [Variable](#) via the [Variables](#) global collection.

### 6.313 rsAggregateInDataRowSortExpression

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsAggregateInDataRowSortExpression* is a critical error that will occur if an expression within [GaugePanel.SortExpressions](#), [Chart.SortExpressions](#), or [Tablix.SortExpressions](#) uses an aggregate function.

### 6.314 rsVariableInDataRegionOrDataSetFilterExpression

*rsVariableInDataRegionOrDataSetFilterExpression* is a critical error that will occur when a [Filter.FilterValues](#) or [Filter.FilterExpression](#) within [Chart.Filters](#), [GaugePanel.Filters](#), [Tablix.Filters](#), or [DataSet.Filters](#) references a [Variable](#) via the [Variables](#) global collection.

### 6.315 rsNestedLookups

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsNestedLookups* is a critical error that will occur when a [Lookup](#), [LookupSet](#), or [MultiLookup](#) function is specified as the argument to another **Lookup**, **LookupSet**, or **MultiLookup** function.

### 6.316 rsLookupInFilterExpression

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsLookupInFilterExpression* is a critical error that will occur when a [Lookup](#), [LookupSet](#), or [MultiLookup](#) function is present within a [Filter.FilterExpression](#) element inside of a [DataSet.Filters](#) element.

### 6.317 rsInvalidLookupScope

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsInvalidLookupScope* is a critical error that will occur when a [Lookup](#), [LookupSet](#), or [MultiLookup](#) function contains as its *Dataset* parameter a value that does not match the value of the **Name** attribute of any DataSet.

### 6.318 rsLookupOfVariable

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

rsLookupOfVariable is a critical error that will occur when a [Lookup](#), [LookupSet](#), or [MultiLookup](#) function contains within any of its arguments a reference to a [Variable](#) via the [Variables](#) global collection.

### 6.319 rsReportItemInLookupDestinationOrResult

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

*rsReportItemInLookupDestinationOrResult* is a critical error that will occur when a [Lookup](#), [LookupSet](#), or [MultiLookup](#) function contains within its *Destination* or *Result* function parameter a reference to a report item via the [ReportItems](#) global collection.

### 6.320 rsAggregateInLookupDestinationOrResult

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

*rsAggregateInLookupDestinationOrResult* is a critical error that will occur when a [Lookup](#), [LookupSet](#), or [MultiLookup](#) function contains an aggregate function within its *Destination* or *Result* function parameter.

### 6.321 rsPagePropertyInSubsequentReportSection

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

*rsPagePropertyInSubsequentReportSection* is a non-critical error that will occur when a [Page](#) element in any [ReportSection](#) except the first **ReportSection** contains any child elements.

### 6.322 rsReportItemReferenceInPageSection

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

*rsReportItemReferenceInPageSection* is a critical error that occurs when an expression in a [PageSection](#) element contains a reference to a report item via the [ReportItems](#) global collection that does not exist in the current [ReportSection](#).

### 6.323 rsInvalidColumnsInReportSection

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

*rsInvalidColumnsInReportSection* is a critical error that will occur when a [Page.Columns](#) element within a [ReportSection](#) is set to a number that is less than 1 or greater than 1000.

### 6.324 rsRowNumberInLookupDestinationOrResult

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

*rsRowNumberInLookupDestinationOrResult* is a critical error that will occur when an expression within the *Destination* or *Result* function parameter of a [Lookup](#), [LookupSet](#), or [MultiLookup](#) function uses a [RowNumber](#) aggregate function.

### 6.325 rsPreviousInLookupDestinationOrResult

**Applies to [RDL 2010/01](#) and [RDL 2016/01](#)**

*rsPreviousInLookupDestinationOrResult* is a critical error that will occur when an expression within the *Destination* or *Result* function parameter of a [Lookup](#), [LookupSet](#), or [MultiLookup](#) function uses a [Previous](#) aggregate function.

### 6.326 rsLevelCallRecursiveHierarchyBothDimensions

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsLevelCallRecursiveHierarchyBothDimensions* is a non-critical error that occurs when an expression contains a [Level](#) function call without a *Scope* parameter set and that expression exists within a scope with recursive dynamic hierarchies in both row and column groups.

For example, if a [TablixCell.CellContents](#) contains the expression "=Level()", and the [Tablix.TablixRowHierarchy](#) and [Tablix.TablixColumnHierarchy](#) both contain recursive [TablixMember](#) elements with [Group.GroupExpressions](#) specified, then this error will occur.

### 6.327 rsInvalidColumnsInBody

**Applies to** [RDL 2003/10](#) and [RDL 2005/01](#)

*rsInvalidColumnsInBody* is a critical error that will occur when the value that is set in the [Body.Columns](#) element is less than 1 or greater than 1000.

### 6.328 rsDuplicateGroupingVariableName

**Applies to** [RDL 2008/01](#), [RDL 2010/01](#), and [RDL 2016/01](#)

*rsDuplicateGroupingVariableName* is a critical error that will occur when a [Variable](#) within a [Group](#) has a non-unique **Name** attribute.

### 6.329 rsNestedCustomAggregate

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsNestedCustomAggregate* is a critical error that will occur when the [Aggregate](#) function is used inside the *Expression* argument of another aggregate function.

### 6.330 rsInvalidNestedAggregateScope

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsInvalidNestedAggregateScope* is a critical error that will occur when an aggregate function that is nested in the *Expression* parameter of another aggregate function specifies a *Scope* parameter that is not the name of a [Group](#) or **DataRegion** contained within the *Scope* parameter that is specified by the outer aggregate function.

### 6.331 rsNestedAggregateScopesFromDifferentAxes

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsNestedAggregateScopesFromDifferentAxes* is a critical error that occurs when an aggregate function references both row and column scopes or scopes from two different axes. For example, an aggregate function can specify a row [Group](#) name as its *Scope* parameter while an aggregate function nested inside its *Expression* parameter can specify a column **Group** name as its *Scope* parameter.

### 6.332 rsIncompatibleNestedAggregateScopes

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsIncompatibleNestedAggregateScopes* is a critical error that will occur when an aggregate function and one or more aggregate functions that are nested within its *Expression* parameter references multiple peer [Groups](#).

### 6.333 rsNestedAggregateScopeRequired

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsNestedAggregateScopeRequired* is a critical error that will occur when the following conditions are true:

- An aggregate function with an omitted *Scope* parameter is nested inside another aggregate function that specifies a *Scope* parameter.
- The expression is located in a cell that belongs to both row and column groups or in a data point that belongs to both series and category groups.

### 6.334 rsInvalidNestedDataSetAggregate

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsInvalidNestedDataSetAggregate* is a critical error that occurs when an aggregate function that is nested inside the *Expression* parameter to another aggregate function has a *Scope* parameter that refers to the name of a dataset.

### 6.335 rsDataSetAggregateOfAggregates

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsDataSetAggregateOfAggregates* is a critical error that will occur when the *Scope* parameter of an aggregate function refers to the name of a data set and the *Expression* parameter of the aggregate function refers to another aggregate function.

### 6.336 rsInvalidNestedRecursiveAggregate

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsInvalidNestedRecursiveAggregate* is a critical error that will occur when an aggregate function is nested inside the *Expression* parameter to another aggregate function and specifies a value for the *Recursive* parameter.

### 6.337 rsRecursiveAggregateOfAggregate

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsRecursiveAggregateOfAggregate* is a critical error that will occur when an aggregate function specifies a value for the *Recursive* parameter and references another aggregate function inside the *Expression* parameter.

### 6.338 rsPostSortAggregateInAggregateExpression

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsPostSortAggregateInAggregateExpression* is a critical error that will occur when an aggregate function references a **First**, **Last**, or **Previous** aggregate function in its *Expression* parameter.

### 6.339 rsRunningValueInAggregateExpression

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsRunningValueInAggregateExpression* is a critical error that will occur when an aggregate function references a [RunningValue](#) aggregate function in its *Expression* parameter.

### 6.340 rsPreviousInAggregateExpression

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsRunningValueInAggregateExpression* is a critical error that will occur when an aggregate function references a [Previous](#) aggregate function in its *Expression* parameter.

### 6.341 rsNestedAggregateViaLookup

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsNestedAggregateViaLookup* is a critical error that will occur when an aggregate function references a [Lookup](#), [LookupSet](#), or [MultiLookup](#) function in its *Expression* parameter and the **Lookup**, **LookupSet**, or **MultiLookup** function references an aggregate function in its *Source* expression parameter.

### 6.342 rsNestedAggregateInPageSection

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsNestedAggregateInPageSection* is a critical error that occurs when an aggregate function located in the page header or page footer refers to another aggregate function in its *Expression* parameter.

### 6.343 rsNestedAggregateInFilterExpression

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsNestedAggregateInFilterExpression* is a critical error that will occur when an aggregate function located in a [Filter](#) expression refers to another aggregate function in its *Expression* parameter.

### 6.344 rsNestedAggregateInGroupVariable

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsNestedAggregateInGroupVariable* is a critical error that will occur when an aggregate function located in a [Group.Variables](#) expression refers to another aggregate function in its *Expression* parameter.

### 6.345 rsVariableTypeNotSerializable

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsVariableTypeNotSerializable* is a critical error that will occur when the type of the value assigned to a [Variable](#) is not serializable.

### 6.346 rsInvalidWritableVariable

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsInvalidWritableVariable* is a critical error that will occur when a [Variable](#) within a [Group.Variables](#) collection has a child [Variable.Writable](#) element that is specified as true.

### 6.347 rsOverallPageNumberInScopedAggregate

*rsOverallPageNumberInScopedAggregate* is a critical error that will occur when an expression within a [PageSection](#) or [PageHeaderFooter](#) contains one of the aggregate functions with its *Scope* parameter set and that expression also contains the "OverallPageNumber" or "OverallTotalPages" variable from the [Globals](#) global collection.

### 6.348 rsOverallPageNumberInBody

*rsOverallPageNumberInBody* is a critical error that will occur when an expression contains the "OverallPageNumber" or "OverallTotalPages" variable from the [Globals](#) global collection and when that expression is not within a [PageSection](#) or [PageHeaderFooter](#) element.

### 6.349 rsParameterPropertyTypeMismatch

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsParameterPropertyTypeMismatch* is a critical error that will occur when a [ReportParameter.DataType](#) element does not match a [Values.Value](#) element that is set within [ReportParameter.DefaultValue](#) or if the **ReportParameter.DataType** element does not match a [ParameterValue.Value](#) element within [ReportParameter.ValidValues](#).

### 6.350 rsStateIndicatorInvalidTransformationScope

**Applies to** [RDL 2010/01](#) and [RDL 2016/01](#)

*rsStateIndicatorInvalidTransformationScope* is a critical error that will occur when a [StateIndicator](#) element has [StateIndicator.TransformationType](#) set to "Percentage" and either **StateIndicator.MinimumValue.Value** or **StateIndicator.MaximumValue.Value** is set to "NaN", but the **StateIndicator** does not have [StateIndicator.TransformationScope](#) specified.

### 6.351 rsVariableInJoinExpression

**Applies to** [RDL 2011/01](#)

*rsVariableInJoinExpression* is a critical error that will occur when [JoinCondition.ForeignKey](#) or [JoinCondition.PrimaryKey](#) contains a reference to a [Variable](#).

### 6.352 rsReportItemInJoinExpression

**Applies to** [RDL 2011/01](#)

*rsReportItemInJoinExpression* is a critical error that will occur when [JoinCondition.ForeignKey](#) or [JoinCondition.PrimaryKey](#) contains a reference to a **ReportItem**.

### 6.353 rsRunningValueInJoinExpression

**Applies to** [RDL 2011/01](#)



*rsRunningValueInJoinExpression* is a critical error that will occur when [JoinCondition.ForeignKey](#) or [JoinCondition.PrimaryKey](#) contains a reference to the [RunningValue](#) aggregate function.

### **6.354 rsPreviousAggregateInJoinExpression**

**Applies to** [RDL 2011/01](#)

*rsPreviousAggregateInJoinExpression* is a critical error that will occur when [JoinCondition.ForeignKey](#) or [JoinCondition.PrimaryKey](#) contains a reference to the [Previous](#) aggregate function.

### **6.355 rsAggregateInJoinExpression**

**Applies to** [RDL 2011/01](#)

*rsAggregateInJoinExpression* is a critical error that will occur when [JoinCondition.ForeignKey](#) or [JoinCondition.PrimaryKey](#) contains a reference to an aggregate function (see section [2.341.7](#)).

### **6.356 rsElementMustContainChildren**

**Applies to** [RDL 2011/01](#)

*rsElementMustContainChildren* is a critical error that will occur when an element is invalid when it does not specify its expected children.

### **6.357 rsElementMustContainChild**

**Applies to** [RDL 2011/01](#)

*rsElementMustContainChild* is a critical error that will occur when an element is a collection of elements and it is invalid when it does not specify at least one child element.

### **6.358 rsMissingDefaultRelationshipJoinCondition**

**Applies to** [RDL 2011/01](#)

*rsMissingDefaultRelationshipJoinCondition* is a critical error that will occur when a [DataSet](#) specifies a [DefaultRelationship](#) for a **RelatedDataSet** that does not contain at least one [JoinCondition](#).

### **6.359 rsNonExistingRelationshipRelatedScope**

**Applies to** [RDL 2011/01](#)

*rsNonExistingRelationshipRelatedScope* is a critical error that will occur when a [DataSet](#) specifies a [DefaultRelationship](#) for a **RelatedDataSet** that does not exist in the [Report](#).

### **6.360 rsInvalidSelfJoinRelationship**

**Applies to** [RDL 2011/01](#)

*rsNonExistingRelationshipRelatedScope* is a critical error that will occur when a [DataSet](#) specifies a [DefaultRelationship](#) for a **RelatedDataSet** where [Dataset.Name](#) is equal to **RelatedDataSet**.

### **6.361 rsInvalidDefaultRelationshipNotNaturalJoin**

**Applies to** [RDL 2011/01](#)

*rsInvalidDefaultRelationshipNotNaturalJoin* is a critical error that will occur when a [DataSet](#) specifies a [DefaultRelationship](#) for a **RelatedDataSet** that is missing **NaturalJoin** or the value for **NaturalJoin** is specified as false.

### 6.362 rsInvalidRelationshipGroupingContainerNotNaturalGroup

*Applies to* [RDL 2011/01](#)

*rsInvalidRelationshipGroupingContainerNotNaturalGroup* is a critical error that will occur when a [Relationship](#) for a **data scope** specifies **NaturalJoin** but the containing data scope does not specify **NaturalGroup**. **NaturalJoin** is only effective if all containing groups specify **NaturalGroup**.

### 6.363 rsInvalidRelationshipContainerNotNaturalJoin

*Applies to* [RDL 2011/01](#)

*rsInvalidRelationshipContainerNotNaturalJoin* is a critical error that will occur when a [Relationship](#) for a **data scope** specifies **NaturalJoin** but the containing **Relationship** does not specify **NaturalJoin**. **NaturalJoin** is only effective if all containing **Relationship** elements specify **NaturalJoin**.

### 6.364 rsInvalidDefaultRelationshipDuplicateRelatedDataset

*Applies to* [RDL 2011/01](#)

*rsInvalidDefaultRelationshipDuplicateRelatedDataset* is a critical error that will occur when a [DefaultRelationship](#) for a [DataSet](#) contains a **RelatedDataSet** that has already been specified in a previous **DefaultRelationship** within the list of [DefaultRelationships](#) in a **DataSet**.

### 6.365 rsInvalidDefaultRelationshipCircularReference

*Applies to* [RDL 2011/01](#)

*rsInvalidDefaultRelationshipCircularReference* is a critical error that occurs when a [DefaultRelationship](#) in a [DataSet](#) contains a circular reference. Circular references within a default relationship are not allowed.

### 6.366 rsInvalidRelationshipDataSetUsedMoreThanOnce

*Applies to* [RDL 2011/01](#)

*rsInvalidRelationshipDataSetUsedMoreThanOnce* is a critical error that will occur when a [DataSet](#) is referenced by two **DataRegions** that both have a [Relationship](#) specified and are within the same top level **DataRegion**.

### 6.367 rsInvalidRelationshipDataSet

*Applies to* [RDL 2011/01](#)

*rsInvalidRelationshipDataSet* is a critical error that will occur when a [DataSet](#) is referenced by two **DataRegions** that both have a [Relationship](#) specified and one is descendant of the other.

### 6.368 rsInvalidNaturalSortContainer

*Applies to* [RDL 2011/01](#)

*rsInvalidNaturalSortContainer* is a critical error that occurs when **NaturalSort** is specified in a scope without the **NaturalGroup** flag set to true in the containing [Group](#).

### 6.369 rsInvalidSortingContainerNotNaturalSort

*Applies to* [RDL 2011/01](#)

*rsInvalidSortingContainerNotNaturalSort* is a critical error that occurs when a grouping specifies **NaturalSort** but the containing scope that is bound to the same [DataSet](#) does not specify **NaturalSort**. **NaturalSort** is effective only if all containing groups and data regions that are bound to the same dataset use **NaturalSort**.

### 6.370 rsConflictingNaturalSortRequirements

*Applies to* [RDL 2011/01](#)

*rsConflictingNaturalSortRequirements* is a critical error that occurs when two or more groupings associated with a [DataSet](#) describe conflicting **NaturalSort** requirements. A grouping without an ancestor/descendent relationship associated with the same dataset can use **NaturalSort** only if they define the same sequence of [SortExpression](#).

### 6.371 rsIncompatibleNaturalSortAndNaturalGroup

*Applies to* [RDL 2011/01](#)

*rsIncompatibleNaturalSortAndNaturalGroup* is a critical error that will occur when grouping specifies both **NaturalGroup** and **NaturalSort** but the [GroupExpressions](#) and [SortExpressions](#) describe conflicting order requirements on the dataset. The **SortExpressions** element for a grouping needs to contain all **GroupExpressions** for the [Group](#) when both **NaturalGroup** and **NaturalSort** are specified.

### 6.372 rsInvalidNaturalSortFlagCombination

*Applies to* [RDL 2011/01](#)

*rsInvalidNaturalSortFlagCombination* is a critical error that will occur when an individual [SortExpression](#) element for a grouping does not share the same value for the **NaturalSort** attribute.

### 6.373 rsInvalidGroupingNaturalGroupFeature

*Applies to* [RDL 2011/01](#)

*rsInvalidGroupingNaturalGroupFeature* is a non-critical error that will occur when a grouping use **NaturalGroup** with **DomainScopes** and **Parent** expressions.

### 6.374 rsInvalidGroupingContainerNotNaturalGroup

*Applies to* [RDL 2011/01](#)

*rsInvalidGroupingContainerNotNaturalGroup* is a non-critical error that will occur when a [Group](#) specifies **NaturalGroup** but the containing **Group** does not use **NaturalGroup**.

### 6.375 rsConflictingNaturalGroupRequirements

*Applies to* [RDL 2011/01](#)

*rsConflictingNaturalGroupRequirements* is a critical error that occurs when two or more groupings associated with a [DataSet](#) describe conflicting **NaturalGroup** requirements. A grouping without an ancestor/descendent relationship associated with the same dataset can use **NaturalGroup** only if they define the same sequence of **GroupExpression**.

### 6.376 rsInvalidBandInvalidLayoutDirection

**Applies to** [RDL 2011/01](#)

*rsInvalidBandInvalidLayoutDirection* is a critical error that will occur when a [Tablix](#) element has [BandLayoutOptions](#) specified and [Tablix.LayoutDirection](#) is not set to "LTR".

### 6.377 rsInvalidBandPageBreakIsSet

**Applies to** [RDL 2011/01](#)

*rsInvalidBandPageBreakIsSet* is a critical error that will occur when a [Tablix](#) element has [BandLayoutOptions](#) specified and the **Tablix** has at least one [Group](#) that has the [Group.PageBreak](#) element specified.

### 6.378 rsInvalidBandShouldNotBeTogglable

**Applies to** [RDL 2011/01](#)

*rsInvalidBandShouldNotBeTogglable* is a critical error that will occur when a [Tablix](#) element has [BandLayoutOptions](#) specified and the **Tablix** has at least one dynamic member with [Visibility.ToggleItem](#) present.

### 6.379 rsInvalidBandNavigationReference

**Applies to** [RDL 2011/01](#)

*rsInvalidBandNavigationReference* is a critical error that occurs when a [NavigationItem.ReportItemReference](#) element is set to a report item that is outside the body of the band's nested tablix or in the grouping of the nested tablix or inside a nested tablix that has the [Tablix.Filters](#) element specified.

### 6.380 rsInvalidBandNavigationItem

**Applies to** [RDL 2011/01](#)

*rsInvalidBandNavigationItem* is a critical error that will occur when a [NavigationItem](#) element has both [NavigationItem.ReportItemReference](#) and [NavigationItem.ReportItem](#) specified.

### 6.381 rsInvalidBandNavigations

**Applies to** [RDL 2011/01](#)

*rsInvalidBandNavigations* is a critical error that will occur when a [BandLayoutOptions](#) element has more than one of the following elements specified:

- [BandLayoutOptions.Coverflow](#)
- [BandLayoutOptions.PlayAxis](#)
- [BandLayoutOptions.Tabstrip](#)

## 6.382 rsInvalidSliderDataSetReference

**Applies to** [RDL 2011/01](#)

*rsInvalidSliderDataSetReference* is a critical error that will occur when a [LabelData.DataSetName](#) child element contains a value that refers to a non-existent [DataSet](#).

## 6.383 rsInvalidSliderDataSetReferenceField

*rsInvalidSliderDataSetReferenceField* is a critical error that will occur when [LabelData.DataSetName](#) has a reference to an existing [DataSet](#), but any of the peer elements (such as [LabelData.Key](#) and [LabelData.Label](#)) has a reference to a non-existing [Field](#) within that **DataSet**.

## 6.384 rsBandKeepTogetherIgnored

**Applies to** [RDL 2011/01](#)

*rsBandKeepTogetherIgnored* is a non-critical error that will occur when a [TablixMember.KeepTogether](#) element is set to true or it is set to an expression for a [Tablix](#) that has [BandLayoutOptions](#) specified.

## 6.385 rsBandIgnoredProperties

**Applies to** [RDL 2011/01](#)

*rsBandIgnoredProperties* is a non-critical error that will occur when [Tablix.BandLayoutOptions](#) is specified and at least one of the following elements are present:

- [Tablix.GroupsBeforeRowHeaders](#)
- [Tablix.RepeatColumnHeaders](#)
- [Tablix.RepeatRowHeaders](#)
- [Tablix.FixedColumnHeaders](#)
- [Tablix.FixedRowHeaders](#)
- [TablixMember.CustomProperties](#)
- [TablixMember.FixedData](#)
- [TablixMember.HideIfNoRows](#)
- [TablixMember.KeepWithGroup](#)
- [TablixMember.RepeatOnNewPage](#)

## 6.386 rsCollationAndCollationCultureSpecified

**Applies to** [RDL 2011/01](#)

*rsCollationAndCollationCultureSpecified* is a critical error that occurs when both [DataSet.Collation](#) and [DataSet.CollationCulture](#) are specified on the same [DataSet](#).

## 6.387 rsInvalidAggregateIndicatorField

**Applies to** [RDL 2011/01](#)

*rsInvalidAggregateIndicatorField* is a critical error that occurs when a [Field.AggregateIndicatorField](#) contains a string that does not match the name of another [Field](#) in the same [DataSet.<196>](#)

### 6.388 rsAggregateIndicatorFieldOnCalculatedField

**Applies to** [RDL 2011/01](#)

*rsAggregateIndicatorFieldOnCalculatedField* is a critical error that occurs when both [Field.AggregateIndicatorField](#) and [Field.Value](#) are specified on the same [Field](#).

### 6.389 rsInvalidSortDirectionMustNotBeSpecified

**Applies to** [RDL 2011/01](#)

*rsInvalidSortDirectionMustNotBeSpecified* is a critical error that occurs when [JoinCondition.SortDirection](#) is specified and **NaturalJoin** is false on the containing [Relationship](#) or [DefaultRelationship](#).

### 6.390 rsInvalidNaturalCrossJoin

**Applies to** [RDL 2011/01](#)

*rsInvalidNaturalCrossJoin* is a critical error that occurs when any of the following is true:

- [Relationship.JoinConditions](#) is not specified and [Relationship.NaturalJoin](#) is false.
- [DefaultRelationship.JoinConditions](#) is not specified and [DefaultRelationship.NaturalJoin](#) is false.
- A [Relationship](#) is not used to correlate a **Group** to its containing **DataRegion** and **Relationship.JoinConditions** is not specified.
- A **DefaultRelationship** is not used to correlate a **Group** to its containing **DataRegion** and **DefaultRelationship.JoinConditions** is not specified.

### 6.391 rsInvalidIntersectionNaturalCrossJoin

**Applies to** [RDL 2011/01](#)

*rsInvalidIntersectionNaturalCrossJoin* is a critical error that occurs when [Relationship.JoinConditions](#) is not specified and the [Relationship](#) is used to correlate an **intersection scope** to a containing group scope or [DefaultRelationship.JoinConditions](#) is not specified and the [DefaultRelationship](#) is used to correlate an intersection scope to a containing group scope.

### 6.392 rsMissingIntersectionRelationshipParentScope

**Applies to** [RDL 2011/01](#)

*rsMissingIntersectionRelationshipParentScope* is a critical error that occurs when a [Relationship](#) is used to correlate an **intersection scope** to a parent scope and [Relationship.ParentScope](#) is not specified or does not contain the name of an immediate parent group scope.

### 6.393 rsInvalidRelationshipDuplicateParentScope

**Applies to** [RDL 2011/01](#)

*rsInvalidRelationshipDuplicateParentScope* is a critical error that occurs when multiple [Relationship](#) elements on the same intersection scope have the same value for [Relationship.ParentScope](#).

### 6.394 rsInvalidCellDataSetName

**Applies to** [RDL 2011/01](#)

*rsInvalidCellDataSetName* is a critical error that occurs when an intersection scope specifies a **DataSetName** ([TablixCell.DataSetName](#), [ChartDataPoint.DataSetName](#), or [DataCell.DataSetName](#)) and both of the following conditions are true:

- No [Relationship](#) exists between the intersection scope and an immediate containing scope.
- No [DefaultRelationship](#) exists between the [DataSet](#) with the specified name and the **DataSet** of an immediate containing scope.

### 6.395 rsDefaultRelationshipIgnored

**Applies to** [RDL 2011/01](#)

*rsDefaultRelationshipIgnored* is a non-critical error that occurs when [DataSet.DefaultRelationships](#) is specified but the [DataSet](#) is not used in any scope in the report.

### 6.396 rsMissingIntersectionDataSetName

**Applies to** [RDL 2011/01](#)

*rsMissingIntersectionDataSetName* is a critical error that occurs when the containing groups of an intersection scope specify different [DataSets](#) and the intersection scope does not specify a **DataSetName** ([TablixCell.DataSetName](#), [ChartDataPoint.DataSetName](#), [DataCell.DataSetName](#).)

### 6.397 rsInvalidRelationshipTopLevelDataRegion

**Applies to** [RDL 2011/01](#)

*rsInvalidRelationshipTopLevelDataRegion* is a critical error that occurs when a **DataRegion** with no containing scope specifies a [Relationship](#).

### 6.398 rsConflictingSortFlags

**Applies to** [RDL 2011/01](#)

*rsConflictingSortFlags* is a critical error that occurs when both [NaturalSort](#) and [DeferredSort](#) are specified and set to true on the same [SortExpression](#).

### 6.399 rsInvalidSortFlagCombination

**Applies to** [RDL 2011/01](#)

*rsInvalidSortFlagCombination* is a critical error that occurs when all individual [SortExpression](#) elements of a [SortExpressions](#) element does not have the same value.

### 6.400 rsInvalidDeferredSortContainer

**Applies to** [RDL 2011/01](#)

*rsInvalidDeferredSortContainer* is a critical error that occurs when [SortExpression.DeferredSort](#) is specified in a [SortExpression](#) that is a child element of [Chart.SortExpressions](#), [CustomData.SortExpressions](#), [GaugePanel.SortExpressions](#), and [Tablix.SortExpressions](#).

## 6.401 rsDuplicateReportSectionName

*Applies to* [RDL 2011/01](#)

*rsDuplicateReportSectionName* is a critical error that occurs when all [ReportSection](#) elements in a [Report](#) do not have [ReportSection.Name](#) with unique values.

## 6.402 rsInvalidFeatureRdlAttribute

*Applies to* [RDL 2011/01](#)

*rsInvalidFeatureRdlAttribute* is a critical error that occurs when an invalid attribute is used in nested elements of the [Report](#).

## 6.403 rsSerializableTypeNotSupported

*Applies to* [RDL 2011/01](#)

*rsSerializableTypeNotSupported* is a critical error that will occur when a **Serializable** type that is not a variant is used in an expression.

## 6.404 rsInvalidScopeReference

*Applies to* [RDL 2012/01](#)

*rsInvalidScopeReference* is a critical error that occurs when there is an expression that has an invalid reference to a [Scope](#) in a **Global.Scopes** collection. Scope references can refer only to a [DataSet](#) that has a [DefaultRelationship](#) to the **DataSet** that is bound to the current scope or the **DataSet** that is bound to a parent scope.

## 6.405 rsInvalidScopeCollectionReference

*Applies to* [RDL 2012/01](#)

*rsInvalidScopeCollectionReference* is a critical error that occurs when there is an expression with a reference to the **Global.Scopes** collection in which it is not allowed. The **Global.Scopes** collection cannot be used outside a **DataRegion**. The **Global.Scopes** collection cannot be used in an aggregate parameter expression.

## 6.406 rsScopeReferenceInComplexExpression

*Applies to* [RDL 2012/01](#)

*rsScopeReferenceInComplexExpression* is a critical error that occurs when a reference to a [Scope](#) in a **Global.Scopes** collection is used in a complex expression. Scope references can be used only in expressions that have the form `Scopes!ScopeName.Fields!FieldName.Value`.

## 6.407 rsScopeReferenceUsesDataSetMoreThanOnce

*Applies to* [RDL 2012/01](#)

*rsScopeReferenceUsesDataSetMoreThanOnce* is a critical error that occurs when all scope references to [DataSet](#) do not originate in the same scope. All references that use the **Global.Scopes** collection to a target **DataSet** need to originate from the same scope.



## 7 Appendix C: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

- Microsoft SQL Server 2000
- Microsoft SQL Server 2005
- Microsoft SQL Server 2008
- Microsoft SQL Server 2008 R2
- Microsoft SQL Server 2012
- Microsoft SQL Server 2014
- Microsoft SQL Server 2016
- Microsoft SQL Server 2017
- Microsoft SQL Server 2019
- Microsoft SQL Server 2022

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> [Section 1.6](#): The following table describes RDL versions and when they were introduced. Newer releases of Microsoft SQL Server Reporting Services support previous RDL versions. For example, Microsoft SQL Server 2022 Reporting Services supports all previous RDL versions.

<b>RDL schema version</b>	<b>Release introduced</b>
<a href="#">RDL 2003/10</a>	Microsoft SQL Server 2000 Reporting Services
<a href="#">RDL 2005/01</a>	Microsoft SQL Server 2005 Reporting Services
<a href="#">RDL 2008/01</a>	Microsoft SQL Server 2008 Reporting Services
<a href="#">RDL 2010/01</a>	Microsoft SQL Server 2008 R2 Reporting Services
<a href="#">RDL 2011/01</a>	Microsoft SQL Server 2012 Reporting Services
<a href="#">RDL 2012/01</a>	Microsoft SQL Server 2012 Service Pack 1 (SP1) Reporting Services
<a href="#">RDL 2013/01</a>	SQL Server 2012 SP1 Reporting Services
<a href="#">RDL 2016/01</a>	Microsoft SQL Server 2016 Reporting Services

<2> [Section 2.2.7](#): The file format validation implementation in SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services restricts sizes to a maximum of 160 inches or the equivalent. Otherwise, Reporting Services restricts sizes to a maximum of 455 inches or the equivalent.

<3> [Section 2.3.7](#): In the Microsoft implementation, the code is specified in a Visual Basic-compatible syntax [MSFT-VBNET].

<4> [Section 2.7.1](#): The value of the [Body.Columns](#) element is less than or equal to 1000.

<5> [Section 2.8.1](#): The value of the **Page.Columns** element is less than or equal to 1000.

<6> [Section 2.18.3](#): The value of the **Paragraph.ListLevel** element is less than or equal to 9.

<7> [Section 2.23.10](#): In the file format validation implementation of **Tablix.DataSetName** in SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services, if **DataRegion** has an ancestor, the value of the **DataSetName** element is ignored. The value of **DataSetName** of the data region's ancestor is used instead.

<8> [Section 2.23.18](#): The file format validation implementation in Reporting Services requires that **Tablix.LayoutDirection** be set to "LTR" if the **Tablix.BandLayoutOptions** element is present. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<9> [Section 2.30.2](#): The uniqueness and content of **TablixCell.DataElementName** is not required for the RDL format but is a Microsoft implementation that is enforced during data rendering to ensure a useable data extract.

<10> [Section 2.30.4](#): The file format validation implementation in Reporting Services requires that **TablixCell.DataSetName** not be specified unless **TablixCell** has both a containing row group and a containing column group. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<11> [Section 2.30.4](#): The file format validation implementation in Reporting Services requires that **TablixCell.DataSetName** does not specify the name of **DataSet** for the containing column group unless the containing row group uses the same **DataSet**. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<12> [Section 2.34.7](#): In Reporting Services, the **TablixMember.KeepTogether** element is automatically set to true if the containing **Tablix** is a band (that is, if the [Tablix.BandLayoutOptions](#) element is specified) and if the **TablixMember.KeepTogether** element is not set to true already. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<13> [Section 2.81.9](#): The file format validation implementation in Reporting Services requires that a grouping does not use **DomainScopes** and **Parent** expressions when **NaturalGroup** is specified. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<14> [Section 2.81.9](#): The file format validation implementation in Reporting Services requires that when **Group** specifies **NaturalGroup**, the containing **Group** specifies **NaturalGroup**. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<15> [Section 2.81.9](#): The file format validation implementation in Reporting Services requires that two or more groupings associated with **DataSet** do not describe conflicting **NaturalGroup** requirements. A grouping without an ancestor/descendent relationship associated with the same dataset can use **NaturalGroup** only if that grouping defines the same sequence of

**GroupExpression**. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<16> [Section 2.81.10](#): The file format validation implementation in Reporting Services requires that **Group.PageBreak** not be specified if the containing **Tablix** is a **band** (that is, if the **Tablix.BandLayoutOptions** element is specified). This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<17> [Section 2.84.2](#): The file format validation implementation in Reporting Services requires that when **NaturalSort** is specified in a scope, the **NaturalGroup** flag for the containing **Group** is specified as true. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<18> [Section 2.84.2](#): The file format validation implementation in Reporting Services requires that when a grouping specifies **NaturalSort**, the containing scope that is bound to the same **DataSet** specifies **NaturalSort**. **NaturalSort** is only effective if all containing groups and data regions bound to the same dataset use **NaturalSort**. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<19> [Section 2.84.2](#): The file format validation implementation in Reporting Services requires that two or more groupings associated with a **DataSet** do not describe conflicting **NaturalSort** requirements. A grouping without an ancestor/descendent relationship that is associated with the same dataset can use **NaturalSort** only if that grouping defines the same sequence of **SortExpression**. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<20> [Section 2.84.2](#): The file format validation implementation in Reporting Services requires that when a grouping specifies both **NaturalGroup** and **NaturalSort**, **GroupExpressions** and **SortExpressions** do not describe conflicting order requirements on the dataset. **SortExpressions** for a grouping has to contain all **GroupExpressions** for the **Group** when both **NaturalGroup** and **NaturalSort** are specified. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<21> [Section 2.84.2](#): The file format validation implementation in Reporting Services requires that the individual **SortExpression** elements in a grouping use the same value for their **NaturalSort** attributes. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<22> [Section 2.88.24](#): In the file format validation implementation for **Chart.DataSetName** in SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services, if **DataRegion** has an ancestor, the value of **DataSetName** element is ignored. The value of **DataSetName** of the data region's ancestor is used instead.

<23> [Section 2.121.2](#): The value of the **ChartAxisScaleBreak.CollapsibleSpaceThreshold** element is greater than or equal to 10 and less than or equal to 90.

<24> [Section 2.121.5](#): The value of the **ChartAxisScaleBreak.MaxNumberOfBreaks** element is less than or equal to 5.

<25> [Section 2.121.6](#): The value of the **ChartAxisScaleBreak.Spacing** element is less than or equal to 10.

<26> [Section 2.128.2](#): The value of the **ChartThreeDProperties.DepthRatio** element is less than or equal to 1000.

<27> [Section 2.128.8](#): The value of the **ChartThreeDProperties.Rotation** element is greater than or equal to -180 and less than or equal to 180.

<28> [Section 2.128.10](#): The value of the **ChartThreeDProperties.WallThickness** element is less than or equal to 30.

<29> [Section 2.141.12](#): The file format validation implementation in Reporting Services requires that **ChartDataPoint.DataSetName** not be specified unless **ChartDataPoint** has both a containing category group and a containing series group. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<30> [Section 2.141.12](#): The file format validation implementation in Reporting Services requires that **ChartDataPoint.DataSetName** does not specify the name of the **DataSet** for the containing series group unless the containing category group uses the same **DataSet**. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<31> [Section 2.146.6](#): The value of the **ChartSmartLabel.CalloutLineWidth** element is greater than or equal to 0.24985pt and less than or equal to 20pt.

<32> [Section 2.152.8](#): The value of the **ChartLegend.ColumnSpacing** element is less than or equal to 100.

<33> [Section 2.163.17](#): In the file format validation implementation for **GaugePanel.DataSetName** in SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services, if a **DataRegion** has an ancestor, the value of the **DataSetName** element is ignored. The value of the **DataSetName** of the data region's ancestor is used instead.

<34> [Section 2.177.3](#): The value of the **CustomLabel.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

<35> [Section 2.178.1](#): The value of the **TickMarkStyle.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

<36> [Section 2.178.5](#): The value of the **TickMarkStyle.Length** element is less than or equal to 100.

<37> [Section 2.178.10](#): The value of the **TickMarkStyle.Width** element is less than or equal to 100.

<38> [Section 2.180.3](#): The value of the **GaugeTickMarks.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

<39> [Section 2.180.7](#): The value of the **GaugeTickMarks.Length** element is less than or equal to 100.

<40> [Section 2.180.12](#): The value of the **GaugeTickMarks.Width** element is less than or equal to 100.

<41> [Section 2.182.4](#): The value of the **LinearPointer.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

<42> [Section 2.182.7](#): The value of the **LinearPointer.MarkerLength** element is less than or equal to 100.

<43> [Section 2.184.1](#): The value of the **Thermometer.BulbOffset** element is less than or equal to 100.

[<44> Section 2.184.2](#): The value of the **Thermometer.BulbSize** element is less than or equal to 1000.

[<45> Section 2.185.2](#): The value of the **ScaleLabels.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

[<46> Section 2.186.1](#): The value of the **ScalePin.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

[<47> Section 2.186.5](#): The value of the **ScalePin.Length** element is less than or equal to 100.

[<48> Section 2.186.10](#): The value of the **ScalePin.Width** element is less than or equal to 100.

[<49> Section 2.187.2](#): The value of the **PinLabel.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

[<50> Section 2.189.4](#): The value of the **ScaleRange.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

[<51> Section 2.189.6](#): The value of the **ScaleRange.EndWidth** element is less than or equal to 1000.

[<52> Section 2.189.13](#): The value of the **ScaleRange.StartWidth** element is less than or equal to 1000.

[<53> Section 2.200.4](#): The value of the **RadialPointer.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

[<54> Section 2.200.7](#): The value of the **RadialPointer.MarkerLength** element is less than or equal to 100.

[<55> Section 2.201.7](#): The value of the **PointerCap.Width** element is less than or equal to 1000.

[<56> Section 2.216.2](#): In the file format validation implementation for **MapDataRegion.DataSetName** in SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services, if **DataRegion** has an ancestor, the value of the **DataSetName** element is ignored. The value of **DataSetName** of the data region's ancestor is used instead.

[<57> Section 2.271.4](#): In the file format validation implementation for **CustomData.DataSetName** in SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services, if **DataRegion** has an ancestor, the value of the **DataSetName** element is ignored. The value of the **DataSetName** of the data region's ancestor is used instead.

[<58> Section 2.278.2](#): The file format validation implementation in Reporting Services requires that **DataCell.DataSetName** not be specified unless **DataCell** has both a containing row group and a containing column group. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

[<59> Section 2.278.2](#): The file format validation implementation in Reporting Services requires that **DataCell.DataSetName** does not specify the name of **DataSet** for the containing column group unless the containing row group uses the same **DataSet**. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

[<60> Section 2.284.4](#): In the Microsoft implementation, the **Top**, **Left**, **Height**, and **Width** child element values of the **ReportItem** element child are interpreted as being those of the **CustomReportItem** element.

<61> [Section 2.310.1](#): Microsoft implementations require this value to be greater than or equal to 1 and less than or equal to 8.

<62> [Section 2.310.2](#): Microsoft implementations require this value to be greater than or equal to 1 and less than or equal to 10000.

<63> [Section 2.311.1](#): Microsoft implementations require the number of consecutive empty rows to be less than or equal to 20.

<64> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Albanian\_100 locale.

<65> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Amharic\_100 locale.

<66> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Arabic\_100 locale.

<67> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Armenian\_100 locale.

<68> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Assamese\_100 locale.

<69> [Section 2.325.4](#): SQL Server 2000 Reporting Services does not support the Azeri\_Cyrillic\_90 locale.

<70> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Azeri\_Cyrillic\_100 locale.

<71> [Section 2.325.4](#): SQL Server 2000 Reporting Services does not support the Azeri\_Latin\_90 locale.

<72> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Azeri\_Latin\_100 locale.

<73> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Bashkir\_100 locale.

<74> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Bengali\_100 locale.

<75> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Bosnian\_Cyrillic\_100 locale.

<76> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Bosnian\_Latin\_100 locale.

<77> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Breton\_100 locale.

<78> [Section 2.325.4](#): SQL Server 2000 Reporting Services does not support the Chinese\_Hong\_Kong\_Stroke\_90 locale.

<79> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese\_Hong\_Kong\_Stroke\_100 locale.

<80> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese\_Macao\_100 locale.



<81> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese\_Macao\_Stroke\_100 locale.

<82> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese\_PRC\_100 locale.

<83> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese\_PRC\_Stroke\_100 locale.

<84> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese\_Simplified\_Pinyin\_100 locale.

<85> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese\_Simplified\_Stroke\_Order\_100 locale.

<86> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese\_Taiwan\_Bopomofo\_100 locale.

<87> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese\_Taiwan\_Stroke\_100 locale.

<88> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese\_Traditional\_Bopomofo\_100 locale.

<89> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese\_Traditional\_Pinyin\_100 locale.

<90> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese\_Traditional\_Stroke\_Count\_100 locale.

<91> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese\_Traditional\_Stroke\_Order\_100 locale.

<92> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Corsican\_100 locale.

<93> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Croatian\_100 locale.

<94> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Cyrillic\_General\_100 locale.

<95> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Czech\_100 locale.

<96> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Danish\_Greenlandic\_100 locale.

<97> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Dari\_100 locale.

<98> [Section 2.325.4](#): SQL Server 2000 Reporting Services does not support the Divehi\_90 locale.

<99> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Divehi\_100 locale.

<100> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Estonian\_100 locale.

<101> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Finnish\_Swedish\_100 locale.

<102> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the French\_100 locale.

<103> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Frisian\_100 locale.

<104> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Georgian\_Traditional\_100 locale.

<105> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the German\_PhoneBook\_100 locale.

<106> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Greek\_100 locale.

<107> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Hebrew\_100 locale.

<108> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Hungarian\_100 locale.

<109> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Hungarian\_Technical\_100 locale.

<110> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Icelandic\_100 locale.

<111> [Section 2.325.4](#): SQL Server 2000 Reporting Services does not support the Indic\_General\_90 locale.

<112> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Indic\_General\_100 locale.

<113> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Inuktitut\_100 locale.

<114> [Section 2.325.4](#): SQL Server 2000 Reporting Services does not support the Japanese\_90 locale.

<115> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Japanese\_100 locale.

<116> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Japanese\_Bushu\_Kakusu\_100 locale.

<117> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Japanese\_Radical\_Stroke\_100 locale.

<118> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Japanese\_Unicode locale.

<119> [Section 2.325.4](#): SQL Server 2000 Reporting Services does not support the Kazakh\_90 locale.

<120> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Kazakh\_100 locale.

<121> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Khmer\_100 locale.

<122> [Section 2.325.4](#): SQL Server 2000 Reporting Services does not support the Korean\_90 locale.



<123> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Korean\_100 locale.

<124> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Lao\_100 locale.

<125> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Latin1\_General\_100 locale.

<126> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Latvian\_100 locale.

<127> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Lithuanian\_100 locale.

<128> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Lithuanian\_Classic locale.

<129> [Section 2.325.4](#): SQL Server 2000 Reporting Services does not support the Macedonian\_FYROM\_90 locale.

<130> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Macedonian\_FYROM\_100 locale.

<131> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Maltese\_100 locale.

<132> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Maori\_100 locale.

<133> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Mapudungan\_100 locale.

<134> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Modern\_Spanish\_100 locale.

<135> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Mohawk\_100 locale.

<136> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Mongolian\_100 locale.

<137> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Nepali\_100 locale.

<138> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Norwegian\_100 locale.

<139> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Norwegian\_Sami\_100 locale.

<140> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Pashto\_100 locale.

<141> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Persian\_100 locale.

<142> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Polish\_100 locale.

<143> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Romanian\_100 locale.

<144> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Romansh\_100 locale.

<145> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Sami\_Norway\_100 locale.

<146> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Sami\_Sweden\_Finland\_100 locale.

<147> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Serbian\_Cyrillic\_100 locale.

<148> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Serbian\_Latin\_100 locale.

<149> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Slovak\_100 locale.

<150> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Slovenian\_100 locale.

<151> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Swedish\_Finnish\_Sami\_100 locale.

<152> [Section 2.325.4](#): SQL Server 2000 Reporting Services does not support the Syriac\_90 locale.

<153> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Syriac\_100 locale.

<154> [Section 2.325.4](#): SQL Server 2000 Reporting Services does not support the Tatar\_90 locale.

<155> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Tatar\_100 locale.

<156> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Tamazight\_100 locale.

<157> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Thai\_100 locale.

<158> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Tibetan\_PRC\_100 locale.

<159> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Traditional\_Spanish\_100 locale.

<160> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Turkish\_100 locale.

<161> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Turkmen\_100 locale.

<162> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Uighur\_PRC\_100 locale.

<163> [Section 2.325.4](#): SQL Server 2000 Reporting Services does not support the Ukrainian\_100 locale.

<164> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Upper\_Sorbian\_100 locale.

<165> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Urdu\_100 locale.

<166> [Section 2.325.4](#): SQL Server 2000 Reporting Services does not support the Uzbek\_Latin\_90 locale.

<167> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Uzbek\_Latin\_100 locale.

<168> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Vietnamese\_100 locale.

<169> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Welsh\_100 locale.

<170> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Yakut\_100 locale.

<171> [Section 2.325.4](#): SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Yi\_100 locale.

<172> [Section 2.325.13](#): In the file format validation implementation in Reporting Services, the behavior of **DateTime** values that specify dates before "March 1, 1900" is undefined when **DataSet.NullsAsBlanks** is true. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<173> [Section 2.328.5](#): In Reporting Services, **Field.AggregateIndicatorField** cannot be present when **Field.Value** is present. If **Field.AggregateIndicatorField** refers to a **Field** in which the **Field.Value** element is present, **Field.Value** on the referenced **Field** has to be a **Boolean** constant, and **Field.Value.DataType** on the referenced **Field** has to be Boolean. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<174> [Section 2.336](#): The file format validation implementation in Reporting Services requires that a **DefaultRelationship** does not contain a circular reference. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<175> [Section 2.336](#): The file format validation implementation in Reporting Services requires that **DefaultRelationship** specifies a **JoinCondition**. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<176> [Section 2.336.1](#): The file format validation implementation in Reporting Services restricts the value of the **DefaultRelationship.RelatedDataSet** element. This element has to be one of the **DataSet.Name** attribute values of the **Report.RelatedDataSet**. **RelatedDataSet** has to be unique among the **RelatedDataSet** elements that are specified for other **DefaultRelationship** elements within the list of **DefaultRelationships** in a dataset. **RelatedDataSet** cannot be equal to the containing **DataSet.Name**. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<177> [Section 2.336.2](#): The file format validation implementation in Reporting Services requires that the **DefaultRelationship.JoinConditions** element is specified unless **DefaultRelationship.NaturalJoin** is true and the **DefaultRelationship** is used to correlate instances of a group scope with its containing **DataRegion**. This behavior does not apply to SQL

Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

[<178> Section 2.336.3](#): The file format validation implementation in Reporting Services requires that the **DefaultRelationship.NaturalJoin** element is specified for **DefaultRelationship** and that the value of the **DefaultRelationship.NaturalJoin** element is not specified as false. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

[<179> Section 2.336.3](#): The file format validation implementation in Reporting Services requires that when two **DataRegions** elements both specify a **Relationship** element and both **DataRegions** elements are within the same top level **DataRegion** element, both **DataRegions** elements cannot reference the same **DataSet** element. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

[<180> Section 2.338.1](#): The file format validation implementation in Reporting Services restricts the content of **JoinCondition.ForeignKey**. This element cannot contain a reference to a **ReportItem** object, a **Variable** element, or a **RunningValue**, **Previous**, or any other aggregate function. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

[<181> Section 2.338.2](#): The file format validation implementation in Reporting Services restricts the content of **JoinCondition.PrimaryKey**. This element cannot contain a reference to a **ReportItem** object, a **Variable** element, or a **RunningValue**, **Previous**, or any other aggregate function. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

[<182> Section 2.340](#): The file format validation implementation in Reporting Services requires that when a **Relationship** element for a **data scope** specifies **NaturalJoin**, the containing **Relationship** has to specify **NaturalJoin**. **NaturalJoin** is effective only if all containing **Relationship** elements specify **NaturalJoin**. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

[<183> Section 2.340](#): The file format validation implementation in Reporting Services requires that when two **DataRegions** both specify a **Relationship** that is within the same top-level **DataRegion** element, both **DataRegions** elements cannot reference the same **DataSet** element. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

[<184> Section 2.340](#): The file format validation implementation in Reporting Services requires that when two **DataRegions** elements specify a **Relationship** element in which one **DataRegions** element is the descendant of the other, both **DataRegions** elements cannot reference the same **DataSet** element. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

[<185> Section 2.340](#): The file format validation implementation in Reporting Services requires that **Relationship** specify **JoinCondition**. If **JoinCondition** is not specified, **DefaultRelationship** of **DataSet** that **ParentScope** is bound to has to specify **JoinCondition**. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

[<186> Section 2.340.2](#): The file format validation implementation in Reporting Services requires that when the **Relationship.NaturalJoin** element is specified for **Relationship**, the containing data scope has to specify **NaturalGroup**. **NaturalJoin** is effective only if all containing groups specify **NaturalGroup**. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server

2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<187> [Section 2.340.3](#): The file format validation implementation in Reporting Services requires that **Relationship.JoinConditions** be specified unless **Relationship.NaturalJoin** is true and **Relationship** is used to correlate instances of a group scope with its containing **DataRegion**. This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

<188> [Section 2.341.3](#): The following namespaces and classes are available:

- **Microsoft.VisualBasic**
- **System.Convert**
- **System.Math**
- **System**

These standard namespaces specify frequently needed functionality as specified in [MSFT-VBNET].

<189> [Section 2.341.4](#): In the Microsoft implementation, the following table specifies how expression return types are converted to RDL data types.

RDL Type	CLR Types
<b>String</b>	<b>String, Char, GUID</b>
<b>Boolean</b>	<b>Boolean</b>
<b>Integer</b>	<b>Int16, Int32, Int64, UInt16, UInt32, UInt64, Byte, SByte, TimeSpan</b> The implementation of Reporting Services can convert <b>Integer</b> expression results to 32-bit or 64-bit precision, depending on the usage context.
<b>DateTime</b>	<b>DateTime, DateTimeOffset</b>
<b>Float</b>	<b>Single, Double, Decimal</b>
<b>Binary</b>	<b>Byte[]</b>
<b>Variant</b>	One of the following RDL types: <b>String, Boolean, Integer, DateTime, Float, or Binary</b> .
<b>VariantArray</b>	Array of <b>Variant</b> . Unless otherwise specified, in the Microsoft implementation, this is a true <b>variant</b> array (that is, <code>object[]</code> ), not a strongly typed array.
<b>StringArray</b>	Array of <b>String</b>
<b>Numeric</b>	One of the following RDL types: <b>Integer</b> or <b>Float</b> .
<b>Scalar</b>	One of the following RDL types: <b>Integer, Float, or DateTime</b> .
<b>Serializable</b>	An RDL <b>Variant</b> , which is a CLR type that is marked with <b>Serializable</b> <a href="#">[ECMA-335]</a> , or a CLR type that implements <b>ISerializable</b> .

<190> [Section 2.341.6.5](#): In the Microsoft implementation, only text box items appear as **ReportItem** objects in the **ReportItems** collection. In third-party implementations, other items can appear in the **ReportItems** collection.

[<191> Section 2.341.6.6](#): The **Name** property is set to "RPL" if the current user request does not use a rendering extension (such as a create history snapshot or toggle event).

The **Name** property is set to NULL for expressions that are evaluated during the data processing phase (for example, grouping, sorting, and filtering).

[<192> Section 2.341.6.6](#): The collection is empty for user requests that do not use a rendering extension (such as a create history snapshot or toggle event) and for expressions that are evaluated during the data processing phase (for example, grouping, sorting, and filtering).

[<193> Section 2.341.6.12](#): The restriction of the use of the **DataSet** and **DataSource** collections in expressions within **Field** elements is not implemented in Reporting Services.

[<194> Section 2.341.7.26.7](#): The Microsoft implementation allows strongly typed arrays in addition to object[].

[<195> Section 2.341.7.26.7](#): The Microsoft implementation allows strongly typed arrays in addition to object[].

[<196> Section 6.387](#): In Reporting Services, *rsInvalidAggregateIndicatorField* also occurs if **Field.AggregateIndicatorField** refers to a field with a **Field.Value** element that does not have a constant **Boolean** value or that does not have **Field.Value.DataType** set to "Boolean". This behavior does not apply to SQL Server 2000 Reporting Services, SQL Server 2005 Reporting Services, SQL Server 2008 Reporting Services, and SQL Server 2008 R2 Reporting Services.

## 8 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

Section	Description	Revision class
<a href="#">2.3</a> Report	Added Report.AuthoringMetadata to the table of Report child elements and added the AuthoringMetadata XSD to the Report XML Schema definition of the Report element.	Major
<a href="#">2.3.2</a> Report.AuthoringMetadata	Added the Report.AuthoringMetadata subsection to the Report section.	Major
<a href="#">2.4</a> AuthoringMetadata	Added the AuthoringMetadata section and 4 subsections (AuthoringMetadata.CreatedBy, AuthoringMetadata.UpdatedBy, AuthoringMetadata.Name, AuthoringMetadata.Version).	Major
<a href="#">5.8</a> RDL XML Schema for Version 2016/01	Added the AuthoringMetadata XSD to the RDL XML Schema for Version 2016/01.	Major



## 9 Index

### A

[abstract base types](#) 64  
[Action element](#) 1112  
[ActionInfo element](#) 1108  
[Aggregate function](#) 1261  
aggregate functions  
  [Aggregate function](#) 1261  
  [Avg function](#) 1255  
  [Count function](#) 1256  
  [CountDistinct function](#) 1256  
  [CountRows function](#) 1257  
  [CreateDrillthroughContext function](#) 1264  
  filters 1262  
  [First function](#) 1258  
  [Last function](#) 1259  
  list of 1252  
  [Max function](#) 1255  
  [Min function](#) 1256  
  [Previous function](#) 1259  
  recursive ([section 2.341.7.2](#) 1253, [section 2.341.7.3](#) 1254, [section 2.341.7.4](#) 1254)  
  restrictions ([section 2.341.7.23](#) 1261, [section 2.341.7.24](#) 1262)  
  [RowNumber function](#) 1260  
  [RunningValue function](#) 1260  
  scope ([section 2.341.7.1](#) 1253, [section 2.341.7.26.1](#) 1263)  
  [StDev function](#) 1257  
  [StDevP function](#) 1257  
  [Sum function](#) 1254  
  [Union function](#) 1258  
  [Var function](#) 1258  
  [VarP function](#) 1258  
[Applicability](#) 71  
[Avg function](#) 1255  
[Axis element](#) 408

### B

[BandLayoutOptions element](#) 263  
[Binary data type](#) 1244  
[Body element](#) 114  
Boolean data type  
  [about](#) 73  
  [expressions](#) 1244  
[Border element](#) 1155  
[BorderStyle element](#) 1159  
[BorderWidth element](#) 1164

### C

[calculations](#) 62  
[CapImage element](#) 830  
[CategoryAxis element](#) 407  
[CategoryGrouping element](#) 419  
[CategoryGroupings element](#) 418  
[CellContents element](#) 234  
[Change tracking](#) 1649  
[Chart element](#) 378  
[chart schema diagram](#) 64

[ChartAlignType element](#) 473  
[ChartAnnotation element](#) 465  
[ChartAnnotations element](#) 465  
[ChartArea element](#) 467  
[ChartAreas element](#) 466  
[ChartAxis element](#) 476  
[ChartCategoryAxes element](#) 475  
[ChartData element](#) 424  
[ChartDataLabel element](#) 548  
[ChartDataPoint element](#) 554  
[ChartDataPoints element](#) 553  
[ChartDataPointValues element](#) 560  
[ChartEmptyPoints element](#) 569  
[ChartHierarchy element](#) 588  
[ChartItemInLegend element](#) 573  
[ChartLegend element](#) 597  
[ChartLegends element](#) 596  
[ChartMarker element](#) 575  
[ChartMember element](#) 591  
[ChartMembers element](#) 590  
[ChartNoMoveDirections element](#) 584  
[ChartSeries element](#) 426  
[ChartSmartLabel element](#) 577  
[Class element](#) 1171  
[Classes element](#) 1170  
[CodeModules element](#) 1172  
[Column and Line Chart example](#) 1281  
[ColumnGrouping element](#) 304  
[ColumnGroupings element](#) 304  
[ConnectionProperties element](#) 1201  
[Corner element](#) 314  
[Count function](#) 1256  
[CountDistinct function](#) 1256  
[CountRows function](#) 1257  
[Coverflow element](#) 266  
[CreateDrillthroughContext function](#) 1264  
custom assemblies  
  [in expressions](#) 1244  
[Custom element](#) 1116  
[custom report item schema diagram](#) 64  
[CustomData element](#) 1074  
[CustomLabel element](#) 707  
[CustomLabels element](#) 706  
[CustomProperties element](#) 1117  
[CustomProperty element](#) 1118  
[CustomReportItem element](#) 1096  
[CustomReportItem example](#) 1302

### D

[Data example](#) 1328  
[data region scope](#) 1253  
data types  
  [Boolean](#) 73  
  [DateTime](#) 74  
  [expressions](#) 1244  
  [Float](#) 74  
  [in expressions](#) 1243  
  [Integer](#) 73  
  [NormalizedString](#) 74  
  [String](#) 73



[DataCell element](#) 1087  
[DataColumnGroupings element](#) 1091  
[DataGrouping element](#) 1092  
[DataGroupings element](#) 1092  
[DataLabel element](#) 441  
[DataPoint element](#) 437  
[DataPoints element](#) 436  
[DataRowGroupings element](#) 1095  
[DataSet element](#) 1204  
[dataset scope](#) 1253  
[DataSetReference element](#) 1188  
[DataSets element](#) 1203  
[DataSource element](#) 1199  
[DataSources element](#) 1198  
[DataValues element](#) 444  
DateTime data type  
  [about](#) 74  
  [expressions](#) 1244  
[DefaultRelationship element](#) 1236  
[DefaultRelationships element](#) 1235  
[DefaultValue element](#) 1187  
[Doughnut Chart example](#) 1299  
[Drillthrough element](#) 1115  
[dynamic scoping](#) 1263  
[DynamicCategories element](#) 420  
[DynamicColumns element](#) 307  
[DynamicRows element](#) 324  
[DynamicSeries element](#) 453

## E

[EmbeddedImage element](#) 1174  
[EmbeddedImages element](#) 1173  
errors  
  [error codes](#) 1584  
  [expression error handling](#) 1244  
[Examples](#) 1268  
  [Column and Line Chart](#) 1281  
  [CustomReportItem](#) 1302  
  [Data](#) 1328  
  [Doughnut Chart](#) 1299  
  [GaugePanel](#) 1332  
  [Line](#) 1338  
  [List](#) 1339  
  [Map](#) 1345  
  [Matrix](#) 1354  
  [Rectangle with Image and Textbox](#) 1343  
  [Report in RDL schema 2005/01](#) 1278  
  [Report in RDL schema 2008/01](#) 1268  
  [Report in RDL schema 2010/01](#) 1272  
  [Subreport](#) 1329  
  [Table](#) 1360  
  [Tablix 1](#) 1307  
  [Tablix 2](#) 1309  
  [Tablix 3](#) 1315  
  [TextBox](#) 1329  
[explicit scope](#) 1253  
Expressions ([section 2.341](#) 1243, [section 2.341.1](#) 1243)  
  data types ([section 2.341.1](#) 1243, [section 2.341.4](#) 1244)  
  [error handling](#) 1244  
  [global collections](#) 1245  
  [referencing custom assemblies](#) 1244  
  [referencing custom code methods](#) 1244

[referencing function libraries](#) 1244  
[syntax](#) 1243

## F

[Field element](#) 1220  
[Fields - vendor-extensible](#) 71  
[Fields element](#) 1219  
[filters and aggregate functions](#) 1262  
[Filters element](#) 1223  
[FilterValues element](#) 1227  
[First function](#) 1258  
Float data type  
  [about](#) 74  
  [expressions](#) 1244  
functions  
  aggregate ([section 2.341.7](#) 1252, [section 2.341.7.23](#) 1261, [section 2.341.7.24](#) 1262)  
  [Aggregate function](#) 1261  
  [Avg function](#) 1255  
  [Count function](#) 1256  
  [CountDistinct function](#) 1256  
  [CountRows function](#) 1257  
  [CreateDrillthroughContext function](#) 1264  
  [First function](#) 1258  
  [InScope function](#) 1263  
  [Last function](#) 1259  
  [Level function](#) 1254  
  [Lookup function](#) 1264  
  [LookupSet function](#) 1264  
  [Max function](#) 1255  
  [Min function](#) 1256  
  [MultiLookup function](#) 1265  
  [Previous function](#) 1259  
  recursive ([section 2.341.7.2](#) 1253, [section 2.341.7.3](#) 1254)  
  [RowNumber function](#) 1260  
  [RunningValue function](#) 1260  
  scope ([section 2.341.7.1](#) 1253, [section 2.341.7.26.1](#) 1263)  
  [semantic queries](#) 1263  
  [StDev function](#) 1257  
  [StDevP function](#) 1257  
  [Sum function](#) 1254  
  [Union function](#) 1258  
  [Var function](#) 1258  
  [VarP function](#) 1258

## G

[gauge panel schema diagram](#) 64  
[GaugeImage element](#) 666  
[GaugeImages element](#) 665  
[GaugeInputValue element](#) 719  
[GaugeLabels element](#) 674  
[GaugeMember element](#) 682  
[GaugePanel element](#) 638  
[GaugePanel example](#) 1332  
[GaugeTickMarks](#) 724  
[Glossary](#) 55  
[Grouping element](#) 370  
[grouping scope](#) 1253

## I

[Image element](#) 134  
[implicit\\_scope](#) 1253  
[IndicatorImage element](#) 859  
[IndicatorState element](#) 851  
[IndicatorStates element](#) 850  
[Informative references](#) 62  
[InScope function](#) 1263  
Integer data type  
  [about](#) 73  
  [expressions](#) 1244  
[Introduction](#) 55

## J

[JoinCondition element](#) 1238  
[JoinConditions element](#) 1237

## L

[LabelData element](#) 272  
[language\\_code](#) 86  
[Last function](#) 1259  
[Level function](#) 1254  
[Line element](#) 146  
[Line example](#) 1338  
[LinearGauge element](#) 685  
[LinearGauges element](#) 684  
[LinearScale element](#) 694  
[LinearScales element](#) 693  
[List element](#) 275  
[List example](#) 1339  
[Localization](#) 71  
[LocIDStringWithDataAttribute type](#) 87  
[Lookup function](#) 1264  
[LookupSet function](#) 1264

## M

[Map example](#) 1345  
[MapBindingFieldPair element](#) 917  
[MapBindingFieldPairs element](#) 916  
[MapBucket element](#) 932  
[MapBuckets element](#) 931  
[MapColorPaletteRule element](#) 924  
[MapColorRangeRule element](#) 933  
[MapColorScale element](#) 878  
[MapColorScaleTitle element](#) 889  
[MapCustomColorRule element](#) 940  
[MapCustomColors elements](#) 947  
[MapCustomView element](#) 1063  
[MapDataBoundView element](#) 1064  
[MapDataRegion element](#) 895  
[MapElementView element](#) 1065  
[MapFieldDefinition element](#) 920  
[MapFieldDefinitions element](#) 919  
[MapFields element](#) 957  
[MapLegend element](#) 1031  
[MapLegends element](#) 1030  
[MapLegendTitle element](#) 1041  
[MapLimits element](#) 1067  
[MapLine element](#) 954  
[MapLineRules element](#) 922  
[MapLines element](#) 953  
[MapLocation element](#) 890  
[MapMarker element](#) 993

[MapMarkerImage element](#) 995  
[MapMarkerRule element](#) 987  
[MapMarkers element](#) 992  
[MapMarkerTemplate element](#) 977  
[MapMeridians element](#) 1069  
[MapParallels element](#) 1072  
[MapPointRules element](#) 984  
[MapPolygonTemplate element](#) 1017  
[MapSizeRule element](#) 947  
[MapTile element](#) 1029  
[MapTileLayer element](#) 1024  
[MapTiles element](#) 1028  
[MapTitle element](#) 1045  
[MapTitles element](#) 1044  
[MapViewport element](#) 1052  
[Marker element](#) 445  
[Matrix example](#) 1354  
[MatrixCell element](#) 320  
[MatrixCells element](#) 319  
[MatrixColumn element](#) 316  
[MatrixColumns element](#) 315  
[MatrixRow element](#) 318  
[MatrixRows element](#) 317  
[Max function](#) 1255  
[Min function](#) 1256  
[MinorGridLines element](#) 417  
[MultiLookup function](#) 1265

## N

[Namespace](#) 63  
[NavigationItem element](#) 270  
[negative RdSize](#) 76  
[NormalizedString data type](#) 74  
[Normative references](#) 61  
[NumericIndicator element](#) 777  
[NumericIndicators element](#) 776

## O

[Overview \(synopsis\)](#) 62

## P

[Page element](#) 117  
[PageHeaderFooter element](#) 126  
[PageSection element](#) 122  
[Paragraph element](#) 189  
[Paragraphs element](#) 188  
[Parameter element](#) 1167  
[Parameters element](#) 1166  
[ParameterValue element](#) 1193  
[ParameterValues element](#) 1192  
[PlayAxis element](#) 268  
[PlotArea element](#) 450  
[PointerCap element](#) 827  
[Previous function](#) 1259  
[Product behavior](#) 1635

## Q

[Query element](#) 1229  
[QueryParameter element](#) 1233  
[QueryParameters element](#) 1232

## R

[RadialPointer element](#) 817  
[RadialPointers element](#) 816  
RDL  
  [abstract base types](#) 64  
  [content](#) 62  
  [report rendering](#) 63  
  [schema diagrams](#) 64  
  [XML file specification](#) 72  
[RdlColor type](#) 78  
[RdlSize type](#) 76  
[RdlURL type](#) 84  
[Rectangle element](#) 152  
[Rectangle with Image and Textbox example](#) 1343  
recursive functions ([section 2.341.7.2](#) 1253, [section 2.341.7.3](#) 1254, [section 2.341.7.4](#) 1254)  
[References](#) 61  
  [informative](#) 62  
  [normative](#) 61  
[Relationship element](#) 1241  
[Relationship to protocols and other structures](#) 70  
[Relationships element](#) 1240  
[rendering reports](#) 63  
report data ([section 1.3.1](#) 62, [section 1.3.4](#) 64)  
[Report element](#) 89  
[Report in RDL schema 2005/01 example](#) 1278  
[Report in RDL schema 2008/01 example](#) 1268  
[Report in RDL schema 2010/01 example](#) 1272  
[report items](#) 64  
report layout ([section 1.3.1](#) 62, [section 1.3.4](#) 64)  
[report rendering](#) 63  
[ReportItems element](#) 128  
[ReportLanguage type](#) 86  
[ReportMIMEType type](#) 84  
[ReportParameters element](#) 1176  
[ReportPath type](#) 85  
[ReportSection element](#) 110  
[ReportSections element](#) 109  
[RowGrouping element](#) 322  
[RowGroupings element](#) 321  
[RowNumber function](#) 1260  
[RunningValue function](#) 1260

## S

[Scalar types](#) 1244  
[ScalePin](#) 753  
[ScaleRange element](#) 765  
[ScaleRanges element](#) 764  
[schema diagrams](#) 64  
scope  
  [about](#) 1253  
  [dynamic scoping](#) 1263  
[semantic queries](#) 1263  
[SeriesGrouping element](#) 452  
[SeriesGroupings element](#) 451  
[Slider element](#) 271  
[SortBy element](#) 377  
[SortExpression element](#) 367  
[SortExpressions element](#) 366  
[Sorting element](#) 376  
[StateImage element](#) 857  
[StateIndicator element](#) 835  
[StateIndicators element](#) 834

[StaticCategories element](#) 422  
[StaticColumn element](#) 313  
[StaticColumns element](#) 312  
[StaticMember element](#) 423  
[StaticRow element](#) 327  
[StaticRows element](#) 326  
[StaticSeries element](#) 455  
[StDev function](#) 1257  
[StDevP function](#) 1257  
String data type  
  [about](#) 73  
  [expressions](#) 1244  
[StringWithDataTypeAttribute type](#) 86  
[Subreport element](#) 164  
[Subreport example](#) 1329  
[Subtotal element](#) 309  
[Sum function](#) 1254

## T

[Table element](#) 328  
[Table example](#) 1360  
[Tablix 1 example](#) 1307  
[Tablix 2 example](#) 1309  
[Tablix 3 example](#) 1315  
[Tablix element](#) 202  
[tablix schema diagram](#) 64  
[TablixBody element](#) 223  
[TablixCell element](#) 230  
[TablixCells element](#) 229  
[TablixColumn element](#) 225  
[TablixColumnHierarchy element](#) 242  
[TablixColumns element](#) 224  
[TablixCorner element](#) 255  
[TablixCornerCell element](#) 259  
[TablixCornerRow element](#) 257  
[TablixCornerRows element](#) 256  
[TablixHeader element](#) 253  
[TablixHierarchy element](#) 260  
[TablixMember element](#) 244  
[TablixMembers element](#) 243  
[TablixRow element](#) 228  
[TablixRowHierarchy element](#) 261  
[TablixRows element](#) 226  
[Tabstrip element](#) 267  
[Textbox element](#) 175  
[TextBox example](#) 1329  
[TextRun element](#) 195  
[TextRuns element](#) 194  
[ThreeDProperties element](#) 455  
[TickMarkStyle element](#) 713  
[Title element](#) 462  
[ToggleImage element](#) 199  
[TopImage element](#) 772  
[Tracking changes](#) 1649  
[transformations](#) 62  
types  
  [Boolean](#) 73  
  [DateTime](#) 74  
  [Float](#) 74  
  [Integer](#) 73  
  [NormalizedString](#) 74  
  [RdlColor](#) 78  
  [RdlSize](#) 76  
  [RdlURL](#) 84

[ReportLanguage](#) 86  
[ReportMimeType](#) 84  
[Reportpath](#) 85  
[String](#) 73

## U

[Union function](#) 1258  
[UserSort element](#) 200

## V

[ValidValues element](#) 1191  
[ValueAxis element](#) 464  
[Values element](#) 1190  
[Var function](#) 1258  
[Variable element](#) 1196  
[Variables element](#) 1195  
[VariantArray](#) 1244  
[VarP function](#) 1258  
[Vendor-extensible fields](#) 71  
[Versioning](#) 71  
[Visibility element](#) 1168

## X

[XML namespace](#) 63  
[XML schema diagrams](#) 64