[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.
- **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.
- **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.

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.

Date	Revision History	Revision Class	Comments
10/16/2019	11.0	Major	Significantly changed the technical content.
12/18/2019	11.1	Minor	Clarified the meaning of the technical content.
6/11/2020	11.2	Minor	Clarified the meaning of the technical content.

Table of Contents

1 Introduction 55
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

2 Structures 72
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 RdlSize 76
2.2.8 RdlColor 78
2.2.9 RdlURL 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 StringWithDataTypeAttribute 86
2.2.15 LocIDStringWithDataTypeAttribute 87
2.2.16 StringWithValueTypeAttribute 88
2.3 Report 89
2.3.1 Report.MustUnderstand 94
2.3.2 Report.Author 94
2.3.3 Report.AutoRefresh 95
2.3.4 Report.Body 95
2.3.5 Report.Classes 95
2.3.6 Report.Code 96
2.3.7 Report.CodeModules 96
2.3.8 Report.ConsumeContainerWhitespace 96
2.3.9 Report.CustomProperties 97
2.3.10 Report.DataElementName 97
2.3.11 Report.DataElementStyle 97
2.3.12 Report.DataSchema 98
2.3.13 Report.DataSets 98
2.3.14 Report.DataSources 99
2.3.15 Report.DataTransform 99
2.3.16 Report.DefaultFontFamily 99
2.3.17 Report.DeferVariableEvaluation 100
2.3.18 Report.Description 100
2.3.19 Report.EmbeddedImages 100

2.3.20 Report.InitialPageName 101
2.3.21 Report.InteractiveHeight 101
2.3.22 Report.InteractiveWidth 101
2.3.23 Report.Language 102
2.3.24 Report.BottomMargin 102
2.3.25 Report.LeftMargin 102
2.3.26 Report.TopMargin 103
2.3.27 Report.RightMargin 103
2.3.28 Report.PageFooter 103
2.3.29 Report.PageHeader 104
2.3.30 Report.PageHeight 104
2.3.31 Report.PageWidth 104
2.3.32 Report.Page 105
2.3.33 Report.ReportParameters 105
2.3.34 Report.ReportParametersLayout 105
2.3.35 Report.Variables 106
2.3.36 Report.Width 106
2.3.37 Report.ReportSections 106
2.4 ReportSections 107
2.4.1 ReportSections.ReportSection 107
2.5 ReportSection 107
2.5.1 ReportSection.Body 109
2.5.2 ReportSection.Page 109
2.5.3 ReportSection.Width 109
2.5.4 ReportSection.DataElementName 110
2.5.5 ReportSection.DataElementOutput 110
2.5.6 ReportSection.Name 111
2.5.7 ReportSection.LayoutDirection 111
2.6 Body 111
2.6.1 Body.Columns 113
2.6.2 Body.ColumnSpacing 113
2.6.3 Body.Style 113
2.6.4 Body.Height 114
2.6.5 Body.ReportItems 114
2.7 Page 114
2.7.1 Page.Columns 116
2.7.2 Page.ColumnSpacing 116
2.7.3 Page.BottomMargin 116
2.7.4 Page.InteractiveHeight 117
2.7.5 Page.InteractiveWidth 117
2.7.6 Page.LeftMargin 117
2.7.7 Page.PageFooter 118
2.7.8 Page.PageHeader 118
2.7.9 Page.PageHeight 118
2.7.10 Page.PageWidth 119
2.7.11 Page.RightMargin 119
2.7.12 Page.Style 119
2.7.13 Page.TopMargin 120
2.8 PageSection 120
2.8.1 PageSection.Style 121
2.8.2 PageSection.Height 122
2.8.3 PageSection.PrintOnFirstPage 122
2.8.4 PageSection.PrintOnLastPage 122
2.8.5 PageSection.PrintBetweenSections 123
2.8.6 PageSection.ReportItems 123
2.9 PageHeaderFooter 123
2.9.1 PageHeaderFooter.Height 124
2.9.2 PageHeaderFooter.PrintOnFirstPage 125

2.9.3 PageHeaderFooter.PrintOnLastPage 125
2.9.4 PageHeaderFooter.ReportItems 125
2.9.5 PageHeaderFooter.Style 126
2.10 ReportItems 126
2.10.1 ReportItems.Chart 128
2.10.2 ReportItems.CustomReportItem 128
2.10.3 ReportItems.GaugePanel 129
2.10.4 ReportItems.Image 129
2.10.5 ReportItems.Line 129
2.10.6 ReportItems.Map 129
2.10.7 ReportItems.Rectangle 130
2.10.8 ReportItems.Subreport 130
2.10.9 ReportItems.Tablix 130
2.10.10 ReportItems.Textbox 131
2.10.11 ReportItems.List 131
2.10.12 ReportItems.Matrix 131
2.10.13 ReportItems.Table 132
2.11 Image 132
2.11.1 Image.Name 134
2.11.2 Image.Style 135
2.11.3 Image.ActionInfo 135
2.11.4 Image.Bookmark 135
2.11.5 Image.CustomProperties 136
2.11.6 Image.DataElementName 136
2.11.7 Image.DataElementOutput 136
2.11.8 Image.DocumentMapLabel 137
2.11.9 Image.Height 137
2.11.10 Image.Left 137
2.11.11 Image.RepeatWith 138
2.11.12 Image.ToolTip 138
2.11.13 Image.Top 138
2.11.14 Image.Visibility 139
2.11.15 Image.Width 139
2.11.16 Image.ZIndex 139
2.11.17 Image.MIMEType 140
2.11.18 Image.Sizing 140
2.11.19 Image.Source 141
2.11.20 Image.Value 141
2.11.21 Image.Tag 142
2.11.22 Image.Tags 142
2.11.23 Image.EmbeddingMode 142
2.12 Line 143
2.12.1 Line.Name 145
2.12.2 Line.Style 145
2.12.3 Line.ActionInfo 145
2.12.4 Line.Bookmark 145
2.12.5 Line.CustomProperties 146
2.12.6 Line.DataElementName 146
2.12.7 Line.DataElementOutput 146
2.12.8 Line.DocumentMapLabel 147
2.12.9 Line.Height 147
2.12.10 Line.Left 148
2.12.11 Line.RepeatWith 148
2.12.12 Line.ToolTip 148
2.12.13 Line.Top 149
2.12.14 Line.Visibility 149
2.12.15 Line.Width 149
2.12.16 Line.ZIndex 150

2.13 Rectangle 150
2.13.1 Rectangle.Name 153
2.13.2 Rectangle.Style 153
2.13.3 Rectangle.ActionInfo 153
2.13.4 Rectangle.Bookmark 154
2.13.5 Rectangle.CustomProperties 154
2.13.6 Rectangle.DataElementName 154
2.13.7 Rectangle.DataElementOutput 155
2.13.8 Rectangle.DocumentMapLabel 155
2.13.9 Rectangle.Height 156
2.13.10 Rectangle.Left 156
2.13.11 Rectangle.RepeatWith 157
2.13.12 Rectangle.ToolTip 157
2.13.13 Rectangle.Top 157
2.13.14 Rectangle.Visibility 158
2.13.15 Rectangle.Width 158
2.13.16 Rectangle.ZIndex 158
2.13.17 Rectangle.KeepTogether 158
2.13.18 Rectangle.LinkToChild 159
2.13.19 Rectangle.OmitBorderOnPageBreak 159
2.13.20 Rectangle.PageBreak 160
2.13.21 Rectangle.ReportItems 160
2.13.22 Rectangle.PageBreakAtEnd 160
2.13.23 Rectangle.PageBreakAtStart 161
2.13.24 Rectangle.PageName 161
2.14 Subreport 161
2.14.1 Subreport.Name 165
2.14.2 Subreport.Style 165
2.14.3 Subreport.ActionInfo 165
2.14.4 Subreport.Bookmark 165
2.14.5 Subreport.CustomProperties 166
2.14.6 Subreport.DataElementName 166
2.14.7 Subreport.DataElementOutput 166
2.14.8 Subreport.DocumentMapLabel 167
2.14.9 Subreport.Height 167
2.14.10 Subreport.Left 168
2.14.11 Subreport.LinkToChild 168
2.14.12 Subreport.RepeatWith 168
2.14.13 Subreport.ToolTip 169
2.14.14 Subreport.Top 169
2.14.15 Subreport.Visibility 169
2.14.16 Subreport.Width 170
2.14.17 Subreport.ZIndex 170
2.14.18 Subreport.KeepTogether 170
2.14.19 Subreport.MergeTransactions 171
2.14.20 Subreport.NoRowsMessage 171
2.14.21 Subreport.OmitBorderOnPageBreak 171
2.14.22 Subreport.Parameters 172
2.14.23 Subreport.ReportName 172
2.14.24 Subreport.NoRows 172
2.15 Textbox 173
2.15.1 Textbox.Name 176
2.15.2 Textbox.Style 177
2.15.3 Textbox.ActionInfo 177
2.15.4 Textbox.Bookmark 177
2.15.5 Textbox.CanScrollVertically 178
2.15.6 Textbox.CustomProperties 178
2.15.7 Textbox.DataElementName 178

2.15.8 Textbox.DataElementOutput 179
2.15.9 Textbox.DocumentMapLabel 179
2.15.10 Textbox.Height 180
2.15.11 Textbox.Left 180
2.15.12 Textbox.RepeatWith 180
2.15.13 Textbox.ToolTip 181
2.15.14 Textbox.Top 181
2.15.15 Textbox.Visibility 181
2.15.16 Textbox.Width 182
2.15.17 Textbox.ZIndex 182
2.15.18 Textbox.CanGrow 182
2.15.19 Textbox.CanShrink 183
2.15.20 Textbox.DataElementStyle 183
2.15.21 Textbox.HideDuplicates 184
2.15.22 Textbox.KeepTogether 184
2.15.23 Textbox.Paragraphs 185
2.15.24 Textbox.ToggleImage 185
2.15.25 Textbox.UserSort 185
2.15.26 Textbox.Value 185
2.16 Paragraphs 186
2.16.1 Paragraphs.Paragraph 186
2.17 Paragraph 187
2.17.1 Paragraph.HangingIndent 188
2.17.2 Paragraph.LeftIndent 189
2.17.3 Paragraph.ListLevel 189
2.17.4 Paragraph.ListStyle 189
2.17.5 Paragraph.RightIndent 190
2.17.6 Paragraph.SpaceAfter 190
2.17.7 Paragraph.SpaceBefore 191
2.17.8 Paragraph.Style 191
2.17.9 Paragraph.TextRuns 191
2.18 TextRuns 192
2.18.1 TextRuns.TextRun 192
2.19 TextRun 193
2.19.1 TextRun.ActionInfo 194
2.19.2 TextRun.Label 194
2.19.3 TextRun.MarkupType 194
2.19.4 TextRun.Style 195
2.19.5 TextRun.ToolTip 195
2.19.6 TextRun.Value 195
2.19.7 TextRun.Value.DataType 196
2.19.8 TextRun.Value.EvaluationMode 196
2.20 ToggleImage 197
2.20.1 ToggleImage.InitialState 197
2.21 UserSort 198
2.21.1 UserSort.SortExpression 199
2.21.2 UserSort.SortExpressionScope 199
2.21.3 UserSort.SortTarget 200
2.22 Tablix 200
2.22.1 Tablix.Name 205
2.22.2 Tablix.ActionInfo 205
2.22.3 Tablix.BandLayoutOptions 205
2.22.4 Tablix.BottomMargin 205
2.22.5 Tablix.Bookmark 206
2.22.6 Tablix.CanScroll 206
2.22.7 Tablix.CustomProperties 206
2.22.8 Tablix.DataElementName 207
2.22.9 Tablix.DataElementOutput 207

2.22.10 Tablix.DataSetName 208
2.22.11 Tablix.DocumentMapLabel 209
2.22.12 Tablix.Filters 209
2.22.13 Tablix.FixedColumnHeaders 209
2.22.14 Tablix.FixedRowHeaders 210
2.22.15 Tablix.GroupsBeforeRowHeaders 210
2.22.16 Tablix.Height 211
2.22.17 Tablix.KeepTogether 211
2.22.18 Tablix.LayoutDirection 212
2.22.19 Tablix.Left 212
2.22.20 Tablix.LeftMargin 212
2.22.21 Tablix.NoRowsMessage 213
2.22.22 Tablix.OmitBorderOnPageBreak 213
2.22.23 Tablix.PageBreak 214
2.22.24 Tablix.PageName 214
2.22.25 Tablix.Relationship 214
2.22.26 Tablix.RepeatColumnHeaders 215
2.22.27 Tablix.RepeatRowHeaders 215
2.22.28 Tablix.RepeatWith 216
2.22.29 Tablix.ReportSlicerState 216
2.22.30 Tablix.RightMargin 216
2.22.31 Tablix.SortExpressions 216
2.22.32 Tablix.Style 217
2.22.33 Tablix.TablixBody 217
2.22.34 Tablix.TablixColumnHierarchy 217
2.22.35 Tablix.TablixCorner 218
2.22.36 Tablix.TablixRowHierarchy 218
2.22.37 Tablix.ToolTip 218
2.22.38 Tablix.Top 219
2.22.39 Tablix.TopMargin 219
2.22.40 Tablix.Visibility 219
2.22.41 Tablix.Width 220
2.22.42 Tablix.ZIndex 220
2.23 TablixBody 221
2.23.1 TablixBody.TablixColumns 221
2.23.2 TablixBody.TablixRows 222
2.24 TablixColumns 222
2.24.1 TablixColumns.TablixColumn 223
2.25 TablixColumn 223
2.25.1 TablixColumn.Width 224
2.26 TablixRows 224
2.26.1 TablixRows.TablixRow 225
2.27 TablixRow 225
2.27.1 TablixRow.Height 226
2.27.2 TablixRow.TablixCells 226
2.28 TablixCells 227
2.28.1 TablixCells.TablixCell 227
2.29 TablixCell 228
2.29.1 TablixCell.CellContents 229
2.29.2 TablixCell.DataElementName 229
2.29.3 TablixCell.DataElementOutput 230
2.29.4 TablixCell.DataSetName 231
2.29.5 TablixCell.Relationships 231
2.30 CellContents 231
2.30.1 CellContents.Chart 233
2.30.2 CellContents.ColSpan 234
2.30.3 CellContents.CustomReportItem 234
2.30.4 CellContents.GaugePanel 235

2.30.5 CellContents.Image 235
2.30.6 CellContents.Line....236
2.30.7 CellContents.Map....236
2.30.8 CellContents.Rectangle....237
2.30.9 CellContents.RowSpan 237
2.30.10 CellContents.Subreport 238
2.30.11 CellContents.Tablix 238
2.30.12 CellContents.Textbox 239
2.31 TablixColumnHierarchy 239
2.31.1 TablixColumnHierarchy.TablixMembers....240
2.31.2 TablixColumnHierarchy.EnableDrilldown 240
2.32 TablixMembers 241
2.32.1 TablixMembers.TablixMember 242
2.33 TablixMember....242
2.33.1 TablixMember.CustomProperties....244
2.33.2 TablixMember.DataElementName 244
2.33.3 TablixMember.DataElementOutput....245
2.33.4 TablixMember.FixedData 246
2.33.5 TablixMember.Group....247
2.33.6 TablixMember.HideIfNoRows....247
2.33.7 TablixMember.KeepTogether....247
2.33.8 TablixMember.KeepWithGroup....248
2.33.9 TablixMember.RepeatOnNewPage 249
2.33.10 TablixMember.SortExpressions....250
2.33.11 TablixMember.TablixHeader 250
2.33.12 TablixMember.TablixMembers 250
2.33.13 TablixMember.Visibility....251
2.34 TablixHeader 251
2.34.1 TablixHeader.CellContents 252
2.34.2 TablixHeader.Size....252
2.35 TablixCorner....253
2.35.1 TablixCorner.TablixCornerRows 253
2.36 TablixCornerRows 254
2.36.1 TablixCornerRows.TablixCornerRow 255
2.37 TablixCornerRow....255
2.37.1 TablixCornerRow.TablixCornerCell....256
2.38 TablixCornerCell....256
2.38.1 TablixCornerCell.CellContents....257
2.39 TablixHierarchy....257
2.39.1 TablixHierarchy.TablixMembers 258
2.40 TablixRowHierarchy....259
2.40.1 TablixRowHierarchy.TablixMembers 260
2.40.2 TablixRowHierarchy.EnableDrilldown 260
2.41 BandLayoutOptions 260
2.41.1 BandLayoutOptions.RowCount....261
2.41.2 BandLayoutOptions.ColumnCount 262
2.41.3 BandLayoutOptions.Coverflow 262
2.41.4 BandLayoutOptions.PlayAxis 263
2.41.5 BandLayoutOptions.Tabstrip 263
2.42 Coverflow....263
2.42.1 Coverflow.NavigationItem....264
2.42.2 Coverflow.Slider 264
2.43 Tabstrip 265
2.43.1 Tabstrip.NavigationItem 265
2.43.2 Tabstrip.Slider....265
2.44 PlayAxis....266
2.44.1 PlayAxis.Slider 266
2.44.2 PlayAxis.DockingOption 267

2.45 NavigationItem 267
2.45.1 NavigationItem.ReportItemReference 268
2.45.2 NavigationItem.ReportItem 268
2.46 Slider 269
2.46.1 Slider.Hidden 269
2.46.2 Slider.LabelData 270
2.47 LabelData 270
2.47.1 LabelData.DataSetName 271
2.47.2 LabelData.Key 271
2.47.3 LabelData.KeyFields 272
2.47.4 LabelData.Label 272
2.48 List 272
2.48.1 List.Name 275
2.48.2 List.Style 275
2.48.3 List.Action 275
2.48.4 List.LinkToChild 276
2.48.5 List.Bookmark 276
2.48.6 List.CustomProperties 276
2.48.7 List.DataElementName 277
2.48.8 List.DataElementOutput 277
2.48.9 List.Label 278
2.48.10 List.Height 278
2.48.11 List.Left 279
2.48.12 List.RepeatWith 279
2.48.13 List.ToolTip 279
2.48.14 List.Top 280
2.48.15 List.Visibility 280
2.48.16 List.Width 280
2.48.17 List.ZIndex 281
2.48.18 List.KeepTogether 281
2.48.19 List.NoRows 282
2.48.20 List.PageBreakAtEnd 282
2.48.21 List.PageBreakAtStart 282
2.48.22 List.DataSetName 283
2.48.23 List.Filters 283
2.48.24 List.DataInstanceElementOutput 283
2.48.25 List.DataInstanceName 284
2.48.26 List.FillPage 285
2.48.27 List.Grouping 285
2.48.28 List.ReportItems 285
2.48.29 List.Sorting 286
2.49 Matrix 286
2.49.1 Matrix.Name 288
2.49.2 Matrix.Style 289
2.49.3 Matrix.Action 289
2.49.4 Matrix.LinkToChild 289
2.49.5 Matrix.Bookmark 290
2.49.6 Matrix.CustomProperties 290
2.49.7 Matrix.DataElementName 290
2.49.8 Matrix.DataElementOutput 291
2.49.9 Matrix.Label 292
2.49.10 Matrix.Height 292
2.49.11 Matrix.Left 293
2.49.12 Matrix.RepeatWith 293
2.49.13 Matrix.ToolTip 293
2.49.14 Matrix.Top 294
2.49.15 Matrix.Visibility 294
2.49.16 Matrix.Width 294

2.49.17 Matrix.ZIndex 295
2.49.18 Matrix.KeepTogether 295
2.49.19 Matrix.NoRows 295
2.49.20 Matrix.PageBreakAtEnd 296
2.49.21 Matrix.PageBreakAtStart 296
2.49.22 Matrix.DataSetName 297
2.49.23 Matrix.Filters 297
2.49.24 Matrix.CellDataElementName 297
2.49.25 Matrix.CellDataElementOutput 298
2.49.26 Matrix.ColumnGroupings 298
2.49.27 Matrix.Corner 299
2.49.28 Matrix.GroupsBeforeRowHeaders 299
2.49.29 Matrix.LayoutDirection 300
2.49.30 Matrix.MatrixColumns 300
2.49.31 Matrix.MatrixRows 300
2.49.32 Matrix.RowGroupings 301
2.50 ColumnGroupings 301
2.50.1 ColumnGroupings.ColumnGrouping 302
2.51 ColumnGrouping 302
2.51.1 ColumnGrouping.DynamicColumns 303
2.51.2 ColumnGrouping.FixedHeader 303
2.51.3 ColumnGrouping.Height 303
2.51.4 ColumnGrouping.StaticColumns 304
2.52 DynamicColumns 304
2.52.1 DynamicColumns.Grouping 305
2.52.2 DynamicColumns.ReportItems 305
2.52.3 DynamicColumns.Sorting 306
2.52.4 DynamicColumns.Subtotal 306
2.52.5 DynamicColumns.Visibility 306
2.53 Subtotal 307
2.53.1 Subtotal.DataElementName 307
2.53.2 Subtotal.DataElementOutput 308
2.53.3 Subtotal.Position 308
2.53.4 Subtotal.ReportItems 309
2.53.5 Subtotal.Style 309
2.54 StaticColumns 310
2.54.1 StaticColumns.StaticColumn 310
2.55 StaticColumn 311
2.55.1 StaticColumn.ReportItems 311
2.56 Corner 312
2.56.1 Corner.ReportItems 312
2.57 MatrixColumns 313
2.57.1 MatrixColumns.MatrixColumn 313
2.58 MatrixColumn 314
2.58.1 MatrixColumn.Width 314
2.59 MatrixRows 314
2.59.1 MatrixRows.MatrixRow 315
2.60 MatrixRow 315
2.60.1 MatrixRow.Height 316
2.60.2 MatrixRow.MatrixCells 316
2.61 MatrixCells 317
2.61.1 MatrixCells.MatrixCell 317
2.62 MatrixCell 317
2.62.1 MatrixCell.ReportItems 318
2.63 RowGroupings 318
2.63.1 RowGroupings.RowGrouping 319
2.64 RowGrouping 319
2.64.1 RowGrouping.DynamicRows 320

2.64.2 RowGrouping.FixedHeader 320
2.64.3 RowGrouping.StaticRows 321
2.64.4 RowGrouping.Width 321
2.65 DynamicRows 321
2.65.1 DynamicRows.Grouping 322
2.65.2 DynamicRows.ReportItems 322
2.65.3 DynamicRows.Sorting 323
2.65.4 DynamicRows.Subtotal 323
2.65.5 DynamicRows.Visibility 324
2.66 StaticRows 324
2.66.1 StaticRows.StaticRow 324
2.67 StaticRow 325
2.67.1 StaticRow.ReportItems 325
2.68 Table 326
2.68.1 Table.Name 328
2.68.2 Table.Style 328
2.68.3 Table.Action 329
2.68.4 Table.LinkToChild 329
2.68.5 Table.Bookmark 329
2.68.6 Table.CustomProperties 330
2.68.7 Table.DataElementName 330
2.68.8 Table.DataElementOutput 330
2.68.9 Table.Label 331
2.68.10 Table.Height 332
2.68.11 Table.Left 332
2.68.12 Table.RepeatWith 332
2.68.13 Table.ToolTip 333
2.68.14 Table.Top 333
2.68.15 Table.Visibility 333
2.68.16 Table.Width 334
2.68.17 Table.ZIndex 334
2.68.18 Table.KeepTogether 334
2.68.19 Table.NoRows 335
2.68.20 Table.PageBreakAtEnd 335
2.68.21 Table.PageBreakAtStart 335
2.68.22 Table.DataSetName 336
2.68.23 Table.Filters 336
2.68.24 Table.DetailDataCollectionName 337
2.68.25 Table.DetailDataElementName 337
2.68.26 Table.DetailDataElementOutput 338
2.68.27 Table.Details 338
2.68.28 Table.FillPage 339
2.68.29 Table.Footer 339
2.68.30 Table.Header 339
2.68.31 Table.TableColumns 340
2.68.32 Table.TableGroups 340
2.69 Details 340
2.69.1 Details.Grouping 341
2.69.2 Details.Sorting 341
2.69.3 Details.TableRows 341
2.69.4 Details.Visibility 342
2.70 TableRows 342
2.70.1 TableRows.TableRow 343
2.71 TableRow 343
2.71.1 TableRow.Height 344
2.71.2 TableRow.TableCells 344
2.71.3 TableRow.Visibility 344
2.72 TableCells 345

2.72.1 TableCells.TableCell 345
2.73 TableCell 345
2.73.1 TableCell.ColSpan 346
2.73.2 TableCell.ReportItems 346
2.74 Footer 347
2.74.1 Footer.RepeatOnNewPage 348
2.74.2 Footer.TableRows 348
2.75 Header 348
2.75.1 Header.FixedHeader 349
2.75.2 Header.RepeatOnNewPage 349
2.75.3 Header.TableRows 350
2.76 TableColumns 350
2.76.1 TableColumns.TableColumn 350
2.77 TableColumn 351
2.77.1 TableColumn.FixedHeader 351
2.77.2 TableColumn.Visibility 352
2.77.3 TableColumn.Width 352
2.78 TableGroups 352
2.78.1 TableGroups.TableGroup 353
2.79 TableGroup 353
2.79.1 TableGroup.Footer 354
2.79.2 TableGroup.Grouping 354
2.79.3 TableGroup.Header 355
2.79.4 TableGroup.Sorting 355
2.79.5 TableGroup.Visibility 355
2.80 Group 356
2.80.1 Group.Name 358
2.80.2 Group.DataElementName 358
2.80.3 Group.DataElementOutput 358
2.80.4 Group.DataSetName 359
2.80.5 Group.DocumentMapLabel 359
2.80.6 Group.DomainScope 360
2.80.7 Group.Filters 360
2.80.8 Group.GroupExpressions 360
2.80.9 Group.NaturalGroup 361
2.80.10 Group.PageBreak 361
2.80.11 Group.PageName 361
2.80.12 Group.Parent 362
2.80.13 Group.ReGroupExpressions 362
2.80.14 Group.Relationship 362
2.80.15 Group.Variables 363
2.81 GroupExpressions 363
2.81.1 GroupExpressions.GroupExpression 364
2.82 SortExpressions 364
2.82.1 SortExpressions.SortExpression 365
2.83 SortExpression 365
2.83.1 SortExpression.Direction 366
2.83.2 SortExpression.NaturalSort 367
2.83.3 SortExpression.Value 367
2.83.4 SortExpression.DeferredSort 368
2.84 Grouping 368
2.84.1 Grouping.Name 369
2.84.2 Grouping.CustomProperties 370
2.84.3 Grouping.DataCollectionName 370
2.84.4 Grouping.DataElementName 371
2.84.5 Grouping.DataElementOutput 371
2.84.6 Grouping.Filters 372
2.84.7 Grouping.GroupExpressions 372

2.84.8 Grouping.Label 372
2.84.9 Grouping.PageBreakAtEnd 373
2.84.10 Grouping.PageBreakAtStart 373
2.84.11 Grouping.Parent 373
2.85 Sorting 374
2.85.1 Sorting.SortBy 375
2.86 SortBy 375
2.86.1 SortBy.Direction 376
2.86.2 SortBy.SortExpression 376
2.87 Chart 376
2.87.1 Chart.Name 383
2.87.2 Chart.Action 383
2.87.3 Chart.ActionInfo 383
2.87.4 Chart.Bookmark 384
2.87.5 Chart.CategoryAxis 384
2.87.6 Chart.CategoryGroupings 384
2.87.7 Chart.ChartAnnotations 385
2.87.8 Chart.ChartAreas 385
2.87.9 Chart.ChartBorderSkin 385
2.87.10 Chart.ChartCodeParameters 385
2.87.11 Chart.ChartCategoryHierarchy 386
2.87.12 Chart.ChartCustomPaletteColors 386
2.87.13 Chart.ChartData 387
2.87.14 Chart.ChartElementOutput 387
2.87.15 Chart.ChartLegends 387
2.87.16 Chart.ChartNoDataMessage 388
2.87.17 Chart.ChartSeriesHierarchy 388
2.87.18 Chart.ChartTitles 388
2.87.19 Chart.Code 389
2.87.20 Chart.CodeLanguage 389
2.87.21 Chart.CustomProperties 389
2.87.22 Chart.DataElementName 390
2.87.23 Chart.DataElementOutput 390
2.87.24 Chart.DataSetName 391
2.87.25 Chart.DocumentMapLabel 391
2.87.26 Chart.DynamicHeight 391
2.87.27 Chart.DynamicWidth 392
2.87.28 Chart.Filters 392
2.87.29 Chart.Height 392
2.87.30 Chart.KeepTogether 393
2.87.31 Chart.Label 393
2.87.32 Chart.Left 393
2.87.33 Chart.Legend 394
2.87.34 Chart.LinkToChild 394
2.87.35 Chart.NoRows 394
2.87.36 Chart.NoRowsMessage 395
2.87.37 Chart.PageBreak 395
2.87.38 Chart.PageBreakAtEnd 395
2.87.39 Chart.PageBreakAtStart 396
2.87.40 Chart.PageName 396
2.87.41 Chart.Palette 396
2.87.42 Chart.PaletteHatchBehavior 398
2.87.43 Chart.PlotArea 398
2.87.44 Chart.PointWidth 398
2.87.45 Chart.Relationship 399
2.87.46 Chart.RepeatWith 399
2.87.47 Chart.SeriesGroupings 400
2.87.48 Chart.SortExpressions 400

2.87.49 Chart.Style 400
2.87.50 Chart.Subtype 400
2.87.51 Chart.ThreeDProperties 401
2.87.52 Chart.Title 402
2.87.53 Chart.ToolTip 402
2.87.54 Chart.Top 402
2.87.55 Chart.Type 403
2.87.56 Chart.ValueAxis 404
2.87.57 Chart.Visibility 404
2.87.58 Chart.Width 404
2.87.59 Chart.ZIndex 404
2.88 CategoryAxis 405
2.88.1 CategoryAxis.Axis 405
2.89 Axis 406
2.89.1 Axis.CrossAt 407
2.89.2 Axis.Interlaced 407
2.89.3 Axis.LogScale 408
2.89.4 Axis.MajorGridLines 408
2.89.5 Axis.MajorInterval 409
2.89.6 Axis.MajorTickMarks 409
2.89.7 Axis.Margin 410
2.89.8 Axis.Max 410
2.89.9 Axis.Min 410
2.89.10 Axis.MinorGridLines 411
2.89.11 Axis.MinorInterval 411
2.89.12 Axis.MinorTickMarks 411
2.89.13 Axis.Reverse 412
2.89.14 Axis.Scalar 412
2.89.15 Axis.Style 413
2.89.16 Axis.Title 413
2.89.17 Axis.Visible 413
2.90 MajorGridLines 414
2.90.1 MajorGridLines.ShowGridLines 414
2.90.2 MajorGridLines.Style 415
2.91 MinorGridLines 415
2.91.1 MinorGridLines.ShowGridLines 415
2.91.2 MinorGridLines.Style 416
2.92 CategoryGroupings 416
2.92.1 CategoryGroupings.CategoryGrouping 417
2.93 CategoryGrouping 417
2.93.1 CategoryGrouping.DynamicCategories 418
2.93.2 CategoryGrouping.StaticCategories 418
2.94 DynamicCategories 418
2.94.1 DynamicCategories.Grouping 419
2.94.2 DynamicCategories.Label 419
2.94.3 DynamicCategories.Sorting 420
2.95 StaticCategories 420
2.95.1 StaticCategories.StaticMember 420
2.96 StaticMember 421
2.96.1 StaticMember.Label 421
2.97 ChartData 422
2.97.1 ChartData.ChartSeries 423
2.97.2 ChartData.ChartSeriesCollection 423
2.97.3 ChartData.ChartDerivedSeriesCollection 423
2.98 ChartSeries 424
2.98.1 ChartSeries.Name 426
2.98.2 ChartSeries.CategoryAxisName 427
2.98.3 ChartSeries.ChartAreaName 427

2.98.4 ChartSeries.ChartDataLabel 427
2.98.5 ChartSeries.ChartDataPoints 428
2.98.6 ChartSeries.ChartEmptyPoints 428
2.98.7 ChartSeries.ChartItemInLegend 428
2.98.8 ChartSeries.ChartMarker 429
2.98.9 ChartSeries.ChartSmartLabel 429
2.98.10 ChartSeries.CustomProperties 429
2.98.11 ChartSeries.DataPoints 430
2.98.12 ChartSeries.Hidden 430
2.98.13 ChartSeries.LegendName 430
2.98.14 ChartSeries.PlotType 431
2.98.15 ChartSeries.Style 431
2.98.16 ChartSeries.Subtype 432
2.98.17 ChartSeries.Type 433
2.98.18 ChartSeries.ValueAxisName 434
2.99 DataPoints 434
2.99.1 DataPoints.DataPoint 435
2.100 DataPoint 435
2.100.1 DataPoint.Action 436
2.100.2 DataPoint.DataElementName 436
2.100.3 DataPoint.DataElementOutput 437
2.100.4 DataPoint.DataLabel 437
2.100.5 DataPoint.DataValues 438
2.100.6 DataPoint.Marker 438
2.100.7 DataPoint.Style 438
2.101 DataLabel 439
2.101.1 DataLabel.Position 440
2.101.2 DataLabel.Rotation 441
2.101.3 DataLabel.Style 441
2.101.4 DataLabel.Value 441
2.101.5 DataLabel.Visible 442
2.102 DataValues 442
2.102.1 DataValues.DataValue 442
2.103 Marker 443
2.103.1 Marker.Size 443
2.103.2 Marker.Style 444
2.103.3 Marker.Type 444
2.104 Legend 445
2.104.1 Legend.InsidePlotArea 446
2.104.2 Legend.Layout 446
2.104.3 Legend.Position 447
2.104.4 Legend.Visible 448
2.105 PlotArea 448
2.105.1 PlotArea.Style 449
2.106 SeriesGroupings 449
2.106.1 SeriesGroupings.SeriesGrouping 449
2.107 SeriesGrouping 450
2.107.1 SeriesGrouping.DynamicSeries 450
2.107.2 SeriesGrouping.StaticSeries 451
2.108 DynamicSeries 451
2.108.1 DynamicSeries.Grouping 452
2.108.2 DynamicSeries.Label 452
2.108.3 DynamicSeries.Sorting 452
2.109 StaticSeries 453
2.109.1 StaticSeries.StaticMember 453
2.110 ThreeDProperties 453
2.110.1 ThreeDProperties.Clustered 455
2.110.2 ThreeDProperties.DepthRatio 455

2.110.3 ThreeDProperties.DrawingStyle 455
2.110.4 ThreeDProperties.Enabled 456
2.110.5 ThreeDProperties.GapDepth 456
2.110.6 ThreeDProperties.HeightRatio 457
2.110.7 ThreeDProperties.Inclination 457
2.110.8 ThreeDProperties.Perspective 457
2.110.9 ThreeDProperties.ProjectionMode 458
2.110.10 ThreeDProperties.Rotation 458
2.110.11 ThreeDProperties.Shading 459
2.110.12 ThreeDProperties.WallThickness 459
2.111 Title 460
2.111.1 Title.Caption 460
2.111.2 Title.Position 461
2.111.3 Title.Style 461
2.112 ValueAxis 462
2.112.1 ValueAxis.Axis 462
2.113 ChartAnnotations 463
2.113.1 ChartAnnotations.ChartAnnotation 463
2.114 ChartAnnotation 463
2.115 ChartAreas 464
2.115.1 ChartAreas.ChartArea 464
2.116 ChartArea 465
2.116.1 ChartArea.Name 466
2.116.2 ChartArea.AlignOrientation 467
2.116.3 ChartArea.AlignWithChartArea 467
2.116.4 ChartArea.ChartAlignType 467
2.116.5 ChartArea.ChartCategoryAxes 468
2.116.6 ChartArea.ChartElementPosition 468
2.116.7 ChartArea.ChartInnerPlotPosition 469
2.116.8 ChartArea.ChartThreeDProperties 469
2.116.9 ChartArea.ChartValueAxes 469
2.116.10 ChartArea.EquallySizedAxesFont 470
2.116.11 ChartArea.Hidden 470
2.116.12 ChartArea.Style 470
2.117 ChartAlignType 471
2.117.1 ChartAlignType.AxesView 471
2.117.2 ChartAlignType.Cursor 472
2.117.3 ChartAlignType.InnerPlotPosition 472
2.117.4 ChartAlignType.Position 473
2.118 ChartCategoryAxes 473
2.118.1 ChartCategoryAxes.ChartAxis 474
2.119 ChartAxis 474
2.119.1 ChartAxis.Name 478
2.119.2 ChartAxis.AllowLabelRotation 478
2.119.3 ChartAxis.Angle 478
2.119.4 ChartAxis.Arrows 479
2.119.5 ChartAxis.ChartAxisScaleBreak 479
2.119.6 ChartAxis.ChartAxisTitle 480
2.119.7 ChartAxis.ChartMajorGridLines 480
2.119.8 ChartAxis.ChartMajorTickMarks 480
2.119.9 ChartAxis.ChartMinorGridLines 481
2.119.10 ChartAxis.ChartMinorTickMarks 481
2.119.11 ChartAxis.ChartStripLines 481
2.119.12 ChartAxis.CrossAt 481
2.119.13 ChartAxis.CustomProperties 482
2.119.14 ChartAxis.HideEndLabels 482
2.119.15 ChartAxis.HideLabels 483
2.119.16 ChartAxis.IncludeZero 483

2.119.17 ChartAxis.Interlaced 483
2.119.18 ChartAxis.InterlacedColor 484
2.119.19 ChartAxis.Interval 484
2.119.20 ChartAxis.IntervalOffset 484
2.119.21 ChartAxis.IntervalOffsetType 485
2.119.22 ChartAxis.IntervalType 485
2.119.23 ChartAxis.LabelInterval 486
2.119.24 ChartAxis.LabelIntervalOffset 487
2.119.25 ChartAxis.LabelIntervalOffsetType 487
2.119.26 ChartAxis.LabelIntervalType 488
2.119.27 ChartAxis.LabelsAutoFitDisabled 488
2.119.28 ChartAxis.Location 489
2.119.29 ChartAxis.LogBase 489
2.119.30 ChartAxis.LogScale 490
2.119.31 ChartAxis.Margin 490
2.119.32 ChartAxis.MarksAlwaysAtPlotEdge 491
2.119.33 ChartAxis.MaxFontSize 491
2.119.34 ChartAxis.Maximum 491
2.119.35 ChartAxis.MinFontSize 492
2.119.36 ChartAxis.Minimum 492
2.119.37 ChartAxis.OffsetLabels 492
2.119.38 ChartAxis.PreventFontGrow 493
2.119.39 ChartAxis.PreventFontShrink 493
2.119.40 ChartAxis.PreventLabelOffset 493
2.119.41 ChartAxis.PreventWordWrap 494
2.119.42 ChartAxis.Reverse 494
2.119.43 ChartAxis.Scalar 494
2.119.44 ChartAxis.Style 495
2.119.45 ChartAxis.VariableAutoInterval 495
2.119.46 ChartAxis.Visible 496
2.120 ChartAxisScaleBreak 496
2.120.1 ChartAxisScaleBreak.BreakLineType 497
2.120.2 ChartAxisScaleBreak.CollapsibleSpaceThreshold 498
2.120.3 ChartAxisScaleBreak.Enabled 498
2.120.4 ChartAxisScaleBreak.IncludeZero 499
2.120.5 ChartAxisScaleBreak.MaxNumberOfBreaks 499
2.120.6 ChartAxisScaleBreak.Spacing 499
2.120.7 ChartAxisScaleBreak.Style 500
2.121 ChartAxisTitle 500
2.121.1 ChartAxisTitle.Caption 501
2.121.2 ChartAxisTitle.Position 501
2.121.3 ChartAxisTitle.Style 502
2.121.4 ChartAxisTitle.TextOrientation 502
2.122 ChartGridLines 503
2.122.1 ChartGridLines.Enabled 504
2.122.2 ChartGridLines.Interval 504
2.122.3 ChartGridLines.IntervalOffset 505
2.122.4 ChartGridLines.IntervalOffsetType 505
2.122.5 ChartGridLines.IntervalType 506
2.122.6 ChartGridLines.Style 507
2.123 ChartStripLines 507
2.123.1 ChartStripLines.ChartStripLine 508
2.124 ChartStripLine 508
2.124.1 ChartStripLine.ActionInfo 509
2.124.2 ChartStripLine.Interval 510
2.124.3 ChartStripLine.IntervalOffset 510
2.124.4 ChartStripLine.IntervalOffsetType 511
2.124.5 ChartStripLine.IntervalType 511

2.124.6 ChartStripLine.StripWidth 512
2.124.7 ChartStripLine.StripWidthType 512
2.124.8 ChartStripLine.Style 513
2.124.9 ChartStripLine.TextOrientation 513
2.124.10 ChartStripLine.Title 514
2.124.11 ChartStripLine.TitleAngle 514
2.124.12 ChartStripLine.ToolTip 515
2.125 ChartTickMarks 515
2.125.1 ChartTickMarks.Enabled 516
2.125.2 ChartTickMarks.Interval 517
2.125.3 ChartTickMarks.IntervalOffset 517
2.125.4 ChartTickMarks.IntervalOffsetType 518
2.125.5 ChartTickMarks.IntervalType 518
2.125.6 ChartTickMarks.Length 519
2.125.7 ChartTickMarks.Style 519
2.125.8 ChartTickMarks.Type 520
2.126 ChartElementPosition 520
2.126.1 ChartElementPosition.Height 521
2.126.2 ChartElementPosition.Left 522
2.126.3 ChartElementPosition.Top 522
2.126.4 ChartElementPosition.Width 522
2.127 ChartThreeDProperties 523
2.127.1 ChartThreeDProperties.Clustered 524
2.127.2 ChartThreeDProperties.DepthRatio 524
2.127.3 ChartThreeDProperties.Enabled 525
2.127.4 ChartThreeDProperties.GapDepth 525
2.127.5 ChartThreeDProperties.Inclination 526
2.127.6 ChartThreeDProperties.Perspective 526
2.127.7 ChartThreeDProperties.ProjectionMode 526
2.127.8 ChartThreeDProperties.Rotation 527
2.127.9 ChartThreeDProperties.Shading 527
2.127.10 ChartThreeDProperties.WallThickness 528
2.128 ChartValueAxes 528
2.128.1 ChartValueAxes.ChartAxis 529
2.129 ChartBorderSkin 529
2.129.1 ChartBorderSkin.ChartBorderSkinType 530
2.129.2 ChartBorderSkin.Style 531
2.130 ChartCodeParameters 531
2.130.1 ChartCodeParameters.ChartCodeParameter 532
2.131 ChartCodeParameter 532
2.131.1 ChartCodeParameter.Name 533
2.131.2 ChartCodeParameter.Value 533
2.132 ChartCustomPaletteColors 533
2.132.1 ChartCustomPaletteColors.ChartCustomPaletteColor 534
2.133 ChartDerivedSeriesCollection 535
2.133.1 ChartDerivedSeriesCollection.ChartDerivedSeries 535
2.134 ChartDerivedSeries 536
2.134.1 ChartDerivedSeries.ChartFormulaParameters 538
2.134.2 ChartDerivedSeries.ChartSeries 538
2.134.3 ChartDerivedSeries.DerivedSeriesFormula 538
2.134.4 ChartDerivedSeries.SourceChartSeriesName 542
2.135 ChartFormulaParameters 542
2.135.1 ChartFormulaParameters.ChartFormulaParameter 543
2.136 ChartFormulaParameter 543
2.136.1 ChartFormulaParameter.Name 544
2.136.2 ChartFormulaParameter.Source 545
2.136.3 ChartFormulaParameter.Value 545
2.137 ChartSeriesCollection 545

2.137.1 ChartSeriesCollection.ChartSeries 546
2.138 ChartDataLabel 546
2.138.1 ChartDataLabel.ActionInfo 547
2.138.2 ChartDataLabel.Label 548
2.138.3 ChartDataLabel.Position 548
2.138.4 ChartDataLabel.Rotation 549
2.138.5 ChartDataLabel.Style 549
2.138.6 ChartDataLabel.ToolTip 550
2.138.7 ChartDataLabel.UseValueAsLabel 550
2.138.8 ChartDataLabel.Visible 550
2.139 ChartDataPoints 551
2.139.1 ChartDataPoints.ChartDataPoint 551
2.140 ChartDataPoint 552
2.140.1 ChartDataPoint.ActionInfo 554
2.140.2 ChartDataPoint.AxisLabel 554
2.140.3 ChartDataPoint.ChartDataLabel 554
2.140.4 ChartDataPoint.ChartDataPointValues 555
2.140.5 ChartDataPoint.ChartItemInLegend 555
2.140.6 ChartDataPoint.ChartMarker 555
2.140.7 ChartDataPoint.CustomProperties 556
2.140.8 ChartDataPoint.DataElementName 556
2.140.9 ChartDataPoint.DataElementOutput 556
2.140.10 ChartDataPoint.Style 557
2.140.11 ChartDataPoint.ToolTip 557
2.140.12 ChartDataPoint.DataSetName 558
2.140.13 ChartDataPoint.Relationships 558
2.141 ChartDataPointValues 558
2.141.1 ChartDataPointValues.End 560
2.141.2 ChartDataPointValues.High 560
2.141.3 ChartDataPointValues.Low 561
2.141.4 ChartDataPointValues.Mean 561
2.141.5 ChartDataPointValues.Median 562
2.141.6 ChartDataPointValues.Size 562
2.141.7 ChartDataPointValues.Start 562
2.141.8 ChartDataPointValues.X 563
2.141.9 ChartDataPointValues.Y 563
2.141.10 ChartDataPointValues.HighlightX 564
2.141.11 ChartDataPointValues.HighlightY 564
2.141.12 ChartDataPointValues.HighlightSize 564
2.141.13 ChartDataPointValues.FormatX 565
2.141.14 ChartDataPointValues.FormatY 565
2.141.15 ChartDataPointValues.FormatSize 566
2.141.16 ChartDataPointValues.CurrencyLanguageX 566
2.141.17 ChartDataPointValues.CurrencyLanguageY 566
2.141.18 ChartDataPointValues.CurrencyLanguageSize 567
2.142 ChartEmptyPoints 567
2.142.1 ChartEmptyPoints.ActionInfo 568
2.142.2 ChartEmptyPoints.AxisLabel 569
2.142.3 ChartEmptyPoints.ChartDataLabel 569
2.142.4 ChartEmptyPoints.ChartMarker 569
2.142.5 ChartEmptyPoints.CustomProperties 570
2.142.6 ChartEmptyPoints.Style 570
2.142.7 ChartEmptyPoints.ToolTip 570
2.143 ChartItemInLegend 571
2.143.1 ChartItemInLegend.ActionInfo 572
2.143.2 ChartItemInLegend.Hidden 572
2.143.3 ChartItemInLegend.LegendText 572
2.143.4 ChartItemInLegend.ToolTip 573

2.144 ChartMarker 573
2.144.1 ChartMarker.Size 574
2.144.2 ChartMarker.Style 574
2.144.3 ChartMarker.Type 575
2.145 ChartSmartLabel 575
2.145.1 ChartSmartLabel.AllowOutSidePlotArea 577
2.145.2 ChartSmartLabel.CalloutBackColor 577
2.145.3 ChartSmartLabel.CalloutLineAnchor 578
2.145.4 ChartSmartLabel.CalloutLineColor 578
2.145.5 ChartSmartLabel.CalloutLineStyle 579
2.145.6 ChartSmartLabel.CalloutLineWidth 579
2.145.7 ChartSmartLabel.CalloutStyle 580
2.145.8 ChartSmartLabel.ChartNoMoveDirections 580
2.145.9 ChartSmartLabel.Disabled 581
2.145.10 ChartSmartLabel.MarkerOverlapping 581
2.145.11 ChartSmartLabel.MaxMovingDistance 581
2.145.12 ChartSmartLabel.MinMovingDistance 582
2.145.13 ChartSmartLabel.ShowOverlapped 582
2.146 ChartNoMoveDirections 582
2.146.1 ChartNoMoveDirections.Down 584
2.146.2 ChartNoMoveDirections.DownLeft 584
2.146.3 ChartNoMoveDirections.DownRight 584
2.146.4 ChartNoMoveDirections.Left 585
2.146.5 ChartNoMoveDirections.Right 585
2.146.6 ChartNoMoveDirections.Up 585
2.146.7 ChartNoMoveDirections.UpLeft 586
2.146.8 ChartNoMoveDirections.UpRight 586
2.147 ChartHierarchy 586
2.147.1 ChartHierarchy.ChartMembers 587
2.147.2 ChartHierarchy.EnableDrilldown 588
2.148 ChartMembers 588
2.148.1 ChartMembers.ChartMember 589
2.149 ChartMember 589
2.149.1 ChartMember.ChartMembers 591
2.149.2 ChartMember.CustomProperties 591
2.149.3 ChartMember.DataElementName 591
2.149.4 ChartMember.DataElementOutput 592
2.149.5 ChartMember.Group 592
2.149.6 ChartMember.Label 593
2.149.7 ChartMember.SortExpressions 593
2.150 ChartLegends 594
2.150.1 ChartLegends.ChartLegend 594
2.151 ChartLegend 595
2.151.1 ChartLegend.Name 597
2.151.2 ChartLegend.AutoFitTextDisabled 597
2.151.3 ChartLegend.ChartElementPosition 597
2.151.4 ChartLegend.ChartLegendColumns 598
2.151.5 ChartLegend.ChartLegendTitle 598
2.151.6 ChartLegend.ColumnSeparator 598
2.151.7 ChartLegend.ColumnSeparatorColor 599
2.151.8 ChartLegend.ColumnSpacing 599
2.151.9 ChartLegend.DockOutsideChartArea 600
2.151.10 ChartLegend.DockToChartArea 600
2.151.11 ChartLegend.EquallySpacedItems 601
2.151.12 ChartLegend.HeaderSeparator 601
2.151.13 ChartLegend.HeaderSeparatorColor 602
2.151.14 ChartLegend.Hidden 602
2.151.15 ChartLegend.InterlacedRows 602

2.151.16 ChartLegend.InterlacedRowsColor....603
2.151.17 ChartLegend.Layout....603
2.151.18 ChartLegend.MaxAutoSize....604
2.151.19 ChartLegend.MinFontSize....604
2.151.20 ChartLegend.Position....604
2.151.21 ChartLegend.Reversed....605
2.151.22 ChartLegend.Style....606
2.151.23 ChartLegend.TextWrapThreshold....606
2.152 ChartLegendColumns....606
2.152.1 ChartLegendColumns.ChartLegendColumn....607
2.153 ChartLegendColumn....607
2.153.1 ChartLegendColumn.Name....609
2.153.2 ChartLegendColumn.ActionInfo....609
2.153.3 ChartLegendColumn.ColumnType....610
2.153.4 ChartLegendColumn.MaximumWidth....610
2.153.5 ChartLegendColumn.MinimumWidth....610
2.153.6 ChartLegendColumn.SeriesSymbolHeight....611
2.153.7 ChartLegendColumn.SeriesSymbolWidth....611
2.153.8 ChartLegendColumn.Style....611
2.153.9 ChartLegendColumn.ToolTip....612
2.153.10 ChartLegendColumn.Value....612
2.154 ChartLegendTitle....612
2.154.1 ChartLegendTitle.Caption....613
2.154.2 ChartLegendTitle.Style....613
2.154.3 ChartLegendTitle.TitleSeparator....614
2.155 ChartTitles....614
2.155.1 ChartTitles.ChartTitle....615
2.156 ChartTitle....615
2.156.1 ChartTitle.Name....617
2.156.2 ChartTitle.ActionInfo....617
2.156.3 ChartTitle.Caption....618
2.156.4 ChartTitle.ChartElementPosition....618
2.156.5 ChartTitle.DockOffset....618
2.156.6 ChartTitle.DockOutsideChartArea....619
2.156.7 ChartTitle.DockToChartArea....619
2.156.8 ChartTitle.Hidden....619
2.156.9 ChartTitle.Position....620
2.156.10 ChartTitle.Style....620
2.156.11 ChartTitle.TextOrientation....621
2.156.12 ChartTitle.ToolTip....621
2.157 ChartLegendColumnHeader....622
2.157.1 ChartLegendColumnHeader.Style....622
2.157.2 ChartLegendColumnHeader.Value....623
2.158 ChartLegendCustomItems....623
2.158.1 ChartLegendCustomItems.ChartLegendCustomItem....624
2.159 ChartLegendCustomItem....624
2.159.1 ChartLegendCustomItem.Name....625
2.159.2 ChartLegendCustomItem.ActionInfo....625
2.159.3 ChartLegendCustomItem.ChartLegendCustomItemCells....626
2.159.4 ChartLegendCustomItem.ChartMarker....626
2.159.5 ChartLegendCustomItem.Separator....626
2.159.6 ChartLegendCustomItem.SeparatorColor....627
2.159.7 ChartLegendCustomItem.Style....627
2.159.8 ChartLegendCustomItem.ToolTip....627
2.160 ChartLegendCustomItemCells....628
2.160.1 ChartLegendCustomItemCells.ChartLegendCustomItemCell....628
2.161 ChartLegendCustomItemCell....629
2.161.1 ChartLegendCustomItemCell.Name....631

2.161.2 ChartLegendCustomItemCell.ActionInfo 631
2.161.3 ChartLegendCustomItemCell.Alignment 631
2.161.4 ChartLegendCustomItemCell.BottomMargin 632
2.161.5 ChartLegendCustomItemCell.CellSpan 632
2.161.6 ChartLegendCustomItemCell.CellType 632
2.161.7 ChartLegendCustomItemCell.ImageHeight 633
2.161.8 ChartLegendCustomItemCell.ImageWidth 633
2.161.9 ChartLegendCustomItemCell.LeftMargin 633
2.161.10 ChartLegendCustomItemCell.RightMargin 634
2.161.11 ChartLegendCustomItemCell.Style 634
2.161.12 ChartLegendCustomItemCell.SymbolHeight 634
2.161.13 ChartLegendCustomItemCell.SymbolWidth 635
2.161.14 ChartLegendCustomItemCell.Text 635
2.161.15 ChartLegendCustomItemCell.ToolTip 635
2.161.16 ChartLegendCustomItemCell.TopMargin 636
2.162 GaugePanel 636
2.162.1 GaugePanel.Name 639
2.162.2 GaugePanel.Style 640
2.162.3 GaugePanel.ActionInfo 640
2.162.4 GaugePanel.Bookmark 640
2.162.5 GaugePanel.CustomProperties 641
2.162.6 GaugePanel.DataElementName 641
2.162.7 GaugePanel.DataElementOutput 641
2.162.8 GaugePanel.DocumentMapLabel 642
2.162.9 GaugePanel.Height 642
2.162.10 GaugePanel.Left 643
2.162.11 GaugePanel.RepeatWith 643
2.162.12 GaugePanel.ToolTip 643
2.162.13 GaugePanel.Top 644
2.162.14 GaugePanel.Visibility 644
2.162.15 GaugePanel.Width 644
2.162.16 GaugePanel.ZIndex 645
2.162.17 GaugePanel.DataSetName 645
2.162.18 GaugePanel.Filters 646
2.162.19 GaugePanel.NoRowsMessage 646
2.162.20 GaugePanel.PageBreak 646
2.162.21 GaugePanel.PageName 647
2.162.22 GaugePanel.SortExpressions 647
2.162.23 GaugePanel.AntiAliasing 647
2.162.24 GaugePanel.AutoLayout 648
2.162.25 GaugePanel.BackFrame 648
2.162.26 GaugePanel.GaugeImages 649
2.162.27 GaugePanel.GaugeLabels 649
2.162.28 GaugePanel.GaugeMember 649
2.162.29 GaugePanel.LinearGauges 650
2.162.30 GaugePanel.NumericIndicators 650
2.162.31 GaugePanel.RadialGauges 650
2.162.32 GaugePanel.ShadowIntensity 651
2.162.33 GaugePanel.StateIndicators 651
2.162.34 GaugePanel.TextAntiAliasingQuality 651
2.162.35 GaugePanel.TopImage 652
2.162.36 GaugePanel.Relationship 652
2.163 BackFrame 653
2.163.1 BackFrame.FrameBackground 654
2.163.2 BackFrame.FrameImage 654
2.163.3 BackFrame.FrameShape 654
2.163.4 BackFrame.FrameStyle 656
2.163.5 BackFrame.FrameWidth 657

2.163.6 BackFrame.GlassEffect 657
2.163.7 BackFrame.Style 658
2.164 FrameBackground 658
2.164.1 FrameBackground.Style 659
2.165 FrameImage 659
2.165.1 FrameImage.MIMEType 660
2.165.2 FrameImage.Source 660
2.165.3 FrameImage.TransparentColor 661
2.165.4 FrameImage.Value 661
2.165.5 FrameImage.ClipImage 662
2.165.6 FrameImage.HueColor 662
2.165.7 FrameImage.Transparency 663
2.166 GaugeImages 663
2.166.1 GaugeImages.GaugeImage 664
2.167 GaugeImage 664
2.167.1 GaugeImage.Name 666
2.167.2 GaugeImage.ActionInfo 666
2.167.3 GaugeImage.Height 666
2.167.4 GaugeImage.Hidden 667
2.167.5 GaugeImage.Left 667
2.167.6 GaugeImage.ParentItem 667
2.167.7 GaugeImage.ToolTip 668
2.167.8 GaugeImage.Top 668
2.167.9 GaugeImage.Width 668
2.167.10 GaugeImage.ZIndex 669
2.167.11 GaugeImage.Angle 669
2.167.12 GaugeImage.MIMEType 669
2.167.13 GaugeImage.ResizeMode 670
2.167.14 GaugeImage.Source 670
2.167.15 GaugeImage.Transparency 671
2.167.16 GaugeImage.TransparentColor 671
2.167.17 GaugeImage.Value 671
2.168 GaugeLabels 672
2.168.1 GaugeLabels.GaugeLabel 672
2.169 GaugeLabel 673
2.169.1 GaugeLabel.Name 674
2.169.2 GaugeLabel.ActionInfo 675
2.169.3 GaugeLabel.Height 675
2.169.4 GaugeLabel.Hidden 675
2.169.5 GaugeLabel.Left 676
2.169.6 GaugeLabel.ParentItem 676
2.169.7 GaugeLabel.ToolTip 677
2.169.8 GaugeLabel.Top 677
2.169.9 GaugeLabel.Width 677
2.169.10 GaugeLabel.ZIndex 678
2.169.11 GaugeLabel.Angle 678
2.169.12 GaugeLabel.ResizeMode 679
2.169.13 GaugeLabel.Style 679
2.169.14 GaugeLabel.Text 679
2.169.15 GaugeLabel.TextShadowOffset 680
2.169.16 GaugeLabel.UseFontPercent 680
2.170 GaugeMember 680
2.170.1 GaugeMember.GaugeMember 681
2.170.2 GaugeMember.Group 682
2.170.3 GaugeMember.SortExpressions 682
2.171 LinearGauges 682
2.171.1 LinearGauges.LinearGauge 683
2.172 LinearGauge 683

2.172.1 LinearGauge.Name 685
2.172.2 LinearGauge.ActionInfo 685
2.172.3 LinearGauge.Height 686
2.172.4 LinearGauge.Hidden 686
2.172.5 LinearGauge.Left 687
2.172.6 LinearGauge.ParentItem 687
2.172.7 LinearGauge.ToolTip 687
2.172.8 LinearGauge.Top 688
2.172.9 LinearGauge.Width 688
2.172.10 LinearGauge.ZIndex 689
2.172.11 LinearGauge.AspectRatio 689
2.172.12 LinearGauge.BackFrame 689
2.172.13 LinearGauge.ClipContent 690
2.172.14 LinearGauge.GaugeScales 690
2.172.15 LinearGauge.TopImage 690
2.172.16 LinearGauge.Orientation 691
2.173 LinearScales 691
2.173.1 LinearScales.LinearScale 692
2.174 LinearScale 692
2.174.1 LinearScale.Name 695
2.174.2 LinearScale.ActionInfo 695
2.174.3 LinearScale.CustomLabels 695
2.174.4 LinearScale.GaugeMajorTickMarks 696
2.174.5 LinearScale.GaugeMinorTickMarks 696
2.174.6 LinearScale.GaugePointers 696
2.174.7 LinearScale.Hidden 697
2.174.8 LinearScale.Interval 697
2.174.9 LinearScale.IntervalOffset 697
2.174.10 LinearScale.Logarithmic 698
2.174.11 LinearScale.LogarithmicBase 698
2.174.12 LinearScale.MaximumPin 699
2.174.13 LinearScale.MaximumValue 699
2.174.14 LinearScale.MinimumPin 699
2.174.15 LinearScale.MinimumValue 700
2.174.16 LinearScale.Multiplier 700
2.174.17 LinearScale.Reversed 701
2.174.18 LinearScale.ScaleLabels 701
2.174.19 LinearScale.ScaleRanges 701
2.174.20 LinearScale.Style 702
2.174.21 LinearScale.TickMarksOnTop 702
2.174.22 LinearScale.ToolTip 702
2.174.23 LinearScale.Width 703
2.174.24 LinearScale.EndMargin 703
2.174.25 LinearScale.Position 703
2.174.26 LinearScale.StartMargin 704
2.175 CustomLabels 704
2.175.1 CustomLabels.CustomLabel 705
2.176 CustomLabel 705
2.176.1 CustomLabel.Name 707
2.176.2 CustomLabel.AllowUpsideDown 707
2.176.3 CustomLabel.DistanceFromScale 707
2.176.4 CustomLabel.FontAngle 708
2.176.5 CustomLabel.Hidden 708
2.176.6 CustomLabel.Placement 709
2.176.7 CustomLabel.RotateLabel 709
2.176.8 CustomLabel.Style 709
2.176.9 CustomLabel.Text 710
2.176.10 CustomLabel.TickMarkStyle 710

2.176.11 CustomLabel.UseFontPercent 710
2.176.12 CustomLabel.Value 711
2.177 TickMarkStyle 711
2.177.1 TickMarkStyle.DistanceFromScale 712
2.177.2 TickMarkStyle.EnableGradient 713
2.177.3 TickMarkStyle.GradientDensity 713
2.177.4 TickMarkStyle.Hidden 714
2.177.5 TickMarkStyle.Length 714
2.177.6 TickMarkStyle.Placement 714
2.177.7 TickMarkStyle.Shape 715
2.177.8 TickMarkStyle.Style 716
2.177.9 TickMarkStyle.TickMarkImage 716
2.177.10 TickMarkStyle.Width 716
2.178 GaugeInputValue 717
2.178.1 GaugeInputValue.AddConstant 718
2.178.2 GaugeInputValue.DataElementName 719
2.178.3 GaugeInputValue.DataElementOutput 719
2.178.4 GaugeInputValue.Formula 720
2.178.5 GaugeInputValue.MaxPercent 720
2.178.6 GaugeInputValue.MinPercent 721
2.178.7 GaugeInputValue.Multiplier 721
2.178.8 GaugeInputValue.Value 722
2.179 GaugeTickMarks 722
2.179.1 GaugeTickMarks.Interval 723
2.179.2 GaugeTickMarks.IntervalOffset 724
2.179.3 GaugeTickMarks.DistanceFromScale 724
2.179.4 GaugeTickMarks.EnableGradient 725
2.179.5 GaugeTickMarks.GradientDensity 725
2.179.6 GaugeTickMarks.Hidden 725
2.179.7 GaugeTickMarks.Length 726
2.179.8 GaugeTickMarks.Placement 726
2.179.9 GaugeTickMarks.Shape 727
2.179.10 GaugeTickMarks.Style 727
2.179.11 GaugeTickMarks.TickMarkImage 728
2.179.12 GaugeTickMarks.Width 728
2.180 LinearPointers 728
2.180.1 LinearPointers.LinearPointer 729
2.181 LinearPointer 730
2.181.1 LinearPointer.Name 731
2.181.2 LinearPointer.ActionInfo 732
2.181.3 LinearPointer.BarStart 732
2.181.4 LinearPointer.DistanceFromScale 733
2.181.5 LinearPointer.GaugeInputValue 733
2.181.6 LinearPointer.Hidden 733
2.181.7 LinearPointer.MarkerLength 734
2.181.8 LinearPointer.MarkerStyle 734
2.181.9 LinearPointer.Placement 735
2.181.10 LinearPointer.PointerImage 735
2.181.11 LinearPointer.SnappingEnabled 735
2.181.12 LinearPointer.SnappingInterval 736
2.181.13 LinearPointer.Style 736
2.181.14 LinearPointer.ToolTip 737
2.181.15 LinearPointer.Width 737
2.181.16 LinearPointer.Thermometer 737
2.181.17 LinearPointer.Type 738
2.182 PointerImage 738
2.182.1 PointerImage.MIMEType 739
2.182.2 PointerImage.Source 740

2.182.3 PointerImage.TransparentColor 740
2.182.4 PointerImage.Value 740
2.182.5 PointerImage.HueColor 741
2.182.6 PointerImage.OffsetX 741
2.182.7 PointerImage.OffsetY 742
2.182.8 PointerImage.Transparency 742
2.183 Thermometer 742
2.183.1 Thermometer.BulbOffset 743
2.183.2 Thermometer.BulbSize 744
2.183.3 Thermometer.Style 744
2.183.4 Thermometer.ThermometerStyle 744
2.184 ScaleLabels 745
2.184.1 ScaleLabels.AllowUpsideDown 746
2.184.2 ScaleLabels.DistanceFromScale 747
2.184.3 ScaleLabels.FontAngle 747
2.184.4 ScaleLabels.Hidden 747
2.184.5 ScaleLabels.Interval 748
2.184.6 ScaleLabels.IntervalOffset 748
2.184.7 ScaleLabels.Placement 749
2.184.8 ScaleLabels.RotateLabels 749
2.184.9 ScaleLabels.ShowEndLabels 749
2.184.10 ScaleLabels.Style 750
2.184.11 ScaleLabels.UseFontPercent 750
2.185 ScalePin 751
2.185.1 ScalePin.DistanceFromScale 752
2.185.2 ScalePin.EnableGradient 753
2.185.3 ScalePin.GradientDensity 753
2.185.4 ScalePin.Hidden 753
2.185.5 ScalePin.Length 754
2.185.6 ScalePin.Placement 754
2.185.7 ScalePin.Shape 755
2.185.8 ScalePin.Style 755
2.185.9 ScalePin.TickMarkImage 756
2.185.10 ScalePin.Width 756
2.185.11 ScalePin.Enable 756
2.185.12 ScalePin.Location 757
2.185.13 ScalePin.PinLabel 757
2.186 PinLabel 757
2.186.1 PinLabel.AllowUpsideDown 759
2.186.2 PinLabel.DistanceFromScale 759
2.186.3 PinLabel.FontAngle 759
2.186.4 PinLabel.Placement 760
2.186.5 PinLabel.RotateLabel 760
2.186.6 PinLabel.Style 761
2.186.7 PinLabel.Text 761
2.186.8 PinLabel.UseFontPercent 761
2.187 ScaleRanges 762
2.187.1 ScaleRanges.ScaleRange 763
2.188 ScaleRange 763
2.188.1 ScaleRange.Name 765
2.188.2 ScaleRange.ActionInfo 765
2.188.3 ScaleRange.BackgroundGradientType 765
2.188.4 ScaleRange.DistanceFromScale 766
2.188.5 ScaleRange.EndValue 766
2.188.6 ScaleRange.EndWidth 767
2.188.7 ScaleRange.Hidden 767
2.188.8 ScaleRange.InRangeBarPointerColor 767
2.188.9 ScaleRange.InRangeLabelColor 768

2.188.10 ScaleRange.InRangeTickMarksColor 768
2.188.11 ScaleRange.Placement 768
2.188.12 ScaleRange.StartValue 769
2.188.13 ScaleRange.StartWidth 769
2.188.14 ScaleRange.Style 770
2.188.15 ScaleRange.ToolTip 770
2.189 TopImage 770
2.189.1 TopImage.MIMEType 772
2.189.2 TopImage.Source 772
2.189.3 TopImage.TransparentColor 772
2.189.4 TopImage.Value 773
2.189.5 TopImage.HueColor 773
2.190 NumericIndicators 774
2.190.1 NumericIndicators.NumericIndicator 774
2.191 NumericIndicator 775
2.191.1 NumericIndicator.Name 777
2.191.2 NumericIndicator.ActionInfo 778
2.191.3 NumericIndicator.Height 778
2.191.4 NumericIndicator.Hidden 778
2.191.5 NumericIndicator.Left 779
2.191.6 NumericIndicator.ParentItem 779
2.191.7 NumericIndicator.ToolTip 779
2.191.8 NumericIndicator.Top 780
2.191.9 NumericIndicator.Width 780
2.191.10 NumericIndicator.ZIndex 780
2.191.11 NumericIndicator.DecimalDigitColor 781
2.191.12 NumericIndicator.DecimalDigits 781
2.191.13 NumericIndicator.DigitColor 781
2.191.14 NumericIndicator.Digits 782
2.191.15 NumericIndicator.GaugeInputValue 782
2.191.16 NumericIndicator.IndicatorStyle 782
2.191.17 NumericIndicator.LedDimColor 783
2.191.18 NumericIndicator.MaximumValue 783
2.191.19 NumericIndicator.MinimumValue 784
2.191.20 NumericIndicator.Multiplier 784
2.191.21 NumericIndicator.NumericIndicatorRanges 784
2.191.22 NumericIndicator.Offstring 785
2.191.23 NumericIndicator.OutOfRangeString 785
2.191.24 NumericIndicator.ResizeMode 785
2.191.25 NumericIndicator.SeparatorColor 785
2.191.26 NumericIndicator.SeparatorWidth 786
2.191.27 NumericIndicator.ShowDecimalPoint 786
2.191.28 NumericIndicator.ShowLeadingZeros 786
2.191.29 NumericIndicator.ShowSign 787
2.191.30 NumericIndicator.SnappingEnabled 787
2.191.31 NumericIndicator.SnappingInterval 788
2.191.32 NumericIndicator.Style 788
2.191.33 NumericIndicator.UseFontPercent 788
2.192 NumericIndicatorRanges 789
2.192.1 NumericIndicatorRanges.NumericIndicatorRange 789
2.193 NumericIndicatorRange 790
2.193.1 NumericIndicatorRange.Name 791
2.193.2 NumericIndicatorRange.StartValue 791
2.193.3 NumericIndicatorRange.EndValue 791
2.193.4 NumericIndicatorRange.DecimalDigitColor 792
2.193.5 NumericIndicatorRange.DigitColor 792
2.194 RadialGauges 792
2.194.1 RadialGauges.RadialGauge 793

2.195 RadialGauge 793
2.195.1 RadialGauge.Name 795
2.195.2 RadialGauge.ActionInfo 796
2.195.3 RadialGauge.Height 796
2.195.4 RadialGauge.Hidden 796
2.195.5 RadialGauge.Left 797
2.195.6 RadialGauge.ParentItem 797
2.195.7 RadialGauge.ToolTip 797
2.195.8 RadialGauge.Top 798
2.195.9 RadialGauge.Width 798
2.195.10 RadialGauge.ZIndex 799
2.195.11 RadialGauge.AspectRatio 799
2.195.12 RadialGauge.BackFrame 799
2.195.13 RadialGauge.ClipContent 800
2.195.14 RadialGauge.GaugeScales 800
2.195.15 RadialGauge.TopImage 800
2.195.16 RadialGauge.PivotX 801
2.195.17 RadialGauge.PivotY 801
2.196 RadialScales 801
2.196.1 RadialScales.RadialScale 802
2.197 RadialScale 802
2.197.1 RadialScale.Name 805
2.197.2 RadialScale.ActionInfo 805
2.197.3 RadialScale.CustomLabels 806
2.197.4 RadialScale.GaugeMajorTickMarks 806
2.197.5 RadialScale.GaugeMinorTickMarks 806
2.197.6 RadialScale.GaugePointers 807
2.197.7 RadialScale.Hidden 807
2.197.8 RadialScale.Interval 807
2.197.9 RadialScale.IntervalOffset 808
2.197.10 RadialScale.Logarithmic 808
2.197.11 RadialScale.LogarithmicBase 808
2.197.12 RadialScale.MaximumPin 809
2.197.13 RadialScale.MaximumValue 809
2.197.14 RadialScale.MinimumPin 810
2.197.15 RadialScale.MinimumValue 810
2.197.16 RadialScale.Multiplier 810
2.197.17 RadialScale.Reversed 811
2.197.18 RadialScale.ScaleLabels 811
2.197.19 RadialScale.ScaleRanges 811
2.197.20 RadialScale.Style 812
2.197.21 RadialScale.TickMarksOnTop 812
2.197.22 RadialScale.ToolTip 812
2.197.23 RadialScale.Width 813
2.197.24 RadialScale.Radius 813
2.197.25 RadialScale.StartAngle 814
2.197.26 RadialScale.SweepAngle 814
2.198 RadialPointers 814
2.198.1 RadialPointers.RadialPointer 815
2.199 RadialPointer 815
2.199.1 RadialPointer.Name 817
2.199.2 RadialPointer.ActionInfo 817
2.199.3 RadialPointer.BarStart 818
2.199.4 RadialPointer.DistanceFromScale 818
2.199.5 RadialPointer.GaugeInputValue 819
2.199.6 RadialPointer.Hidden 819
2.199.7 RadialPointer.MarkerLength 819
2.199.8 RadialPointer.MarkerStyle 820

2.199.9 RadialPointer.Placement 820
2.199.10 RadialPointer.PointerImage 821
2.199.11 RadialPointer.SnappingEnabled 821
2.199.12 RadialPointer.SnappingInterval 821
2.199.13 RadialPointer.Style 822
2.199.14 RadialPointer.ToolTip 822
2.199.15 RadialPointer.Width 823
2.199.16 RadialPointer.NeedleStyle 823
2.199.17 RadialPointer.PointerCap 824
2.199.18 RadialPointer.Type 824
2.200 PointerCap 825
2.200.1 PointerCap.CapImage 826
2.200.2 PointerCap.CapStyle 826
2.200.3 PointerCap.Hidden 827
2.200.4 PointerCap.OnTop 827
2.200.5 PointerCap.Reflection 827
2.200.6 PointerCap.Style 828
2.200.7 PointerCap.Width 828
2.201 CapImage 828
2.201.1 CapImage.MIMEType 830
2.201.2 CapImage.Source 830
2.201.3 CapImage.TransparentColor 830
2.201.4 CapImage.Value 831
2.201.5 CapImage.HueColor 831
2.201.6 CapImage.OffsetX 832
2.201.7 CapImage.OffsetY 832
2.202 StateIndicators 832
2.202.1 StateIndicators.StateIndicator 833
2.203 StateIndicator 833
2.203.1 StateIndicator.Name 836
2.203.2 StateIndicator.ActionInfo 836
2.203.3 StateIndicator.Height 836
2.203.4 StateIndicator.Hidden 837
2.203.5 StateIndicator.Left 837
2.203.6 StateIndicator.MaximumValue 838
2.203.7 StateIndicator.MinimumValue 838
2.203.8 StateIndicator.ParentItem 838
2.203.9 StateIndicator.ScaleFactor 839
2.203.10 StateIndicator.StateDataElementName 839
2.203.11 StateIndicator.StateDataElementOutput 839
2.203.12 StateIndicator.ToolTip 840
2.203.13 StateIndicator.Top 840
2.203.14 StateIndicator.TransformationScope 841
2.203.15 StateIndicator.TransformationType 841
2.203.16 StateIndicator.Width 842
2.203.17 StateIndicator.ZIndex 842
2.203.18 StateIndicator.Angle 843
2.203.19 StateIndicator.GaugeInputValue 843
2.203.20 StateIndicator.IndicatorStates 843
2.203.21 StateIndicator.IndicatorStyle 844
2.203.22 StateIndicator.ResizeMode 846
2.203.23 StateIndicator.IndicatorImage 846
2.203.24 StateIndicator.StateImage 846
2.203.25 StateIndicator.Style 847
2.203.26 StateIndicator.Text 847
2.203.27 StateIndicator.UseFontPercent 847
2.204 IndicatorStates 848
2.204.1 IndicatorStates.IndicatorState 848

2.205 IndicatorState 849
2.205.1 IndicatorState.Name 850
2.205.2 IndicatorState.Color 850
2.205.3 IndicatorState.IndicatorStyle 851
2.205.4 IndicatorState.Style 853
2.205.5 IndicatorState.StartValue 853
2.205.6 IndicatorState.EndValue 853
2.205.7 IndicatorState.IndicatorImage 854
2.205.8 IndicatorState.ScaleFactor 854
2.205.9 IndicatorState.StateImage 854
2.205.10 IndicatorState.Text 855
2.206 StateImage 855
2.206.1 StateImage.MIMEType 856
2.206.2 StateImage.Source 856
2.206.3 StateImage.TransparentColor 856
2.206.4 StateImage.Value 857
2.206.5 StateImage.HueColor 857
2.206.6 StateImage.Transparency 857
2.207 IndicatorImage 857
2.207.1 IndicatorImage.MIMEType 858
2.207.2 IndicatorImage.Source 859
2.207.3 IndicatorImage.TransparentColor 859
2.207.4 IndicatorImage.Value 859
2.207.5 IndicatorImage.HueColor 860
2.207.6 IndicatorImage.Transparency 860
2.208 Map 861
2.208.1 Map.Name 863
2.208.2 Map.ActionInfo 863
2.208.3 Map.AntiAliasing 864
2.208.4 Map.Bookmark 864
2.208.5 Map.CustomProperties 864
2.208.6 Map.DataElementName 865
2.208.7 Map.DataElementOutput 865
2.208.8 Map.DocumentMapLabel 866
2.208.9 Map.Height 866
2.208.10 Map.Left 866
2.208.11 Map.MapBorderSkin 867
2.208.12 Map.MapColorScale 867
2.208.13 Map.MapDataRegions 867
2.208.14 Map.MapDistanceScale 868
2.208.15 Map.MapLayers 868
2.208.16 Map.MapLegends 868
2.208.17 Map.MapTitles 869
2.208.18 Map.MapViewport 869
2.208.19 Map.MaximumSpatialElementCount 869
2.208.20 Map.MaximumTotalPointCount 870
2.208.21 Map.PageBreak 870
2.208.22 Map.PageName 870
2.208.23 Map.RepeatWith 871
2.208.24 Map.ShadowIntensity 871
2.208.25 Map.Style 871
2.208.26 Map.TextAntiAliasingQuality 872
2.208.27 Map.TileLanguage 872
2.208.28 Map.ToolTip 873
2.208.29 Map.Top 873
2.208.30 Map.Visibility 874
2.208.31 Map.Width 874
2.208.32 Map.ZIndex 874

2.209 MapBorderSkin 875
2.209.1 MapBorderSkin.MapBorderSkinType 875
2.209.2 MapBorderSkin.Style 876
2.210 MapColorScale 876
2.210.1 MapColorScale.ActionInfo 878
2.210.2 MapColorScale.BottomMargin 878
2.210.3 MapColorScale.ColorBarBorderColor 879
2.210.4 MapColorScale.DockOutsideViewport 879
2.210.5 MapColorScale.Hidden 879
2.210.6 MapColorScale.HideEndLabels 880
2.210.7 MapColorScale.LabelBehavior 880
2.210.8 MapColorScale.LabelFormat 881
2.210.9 MapColorScale.LabelInterval 881
2.210.10 MapColorScale.LabelPlacement 881
2.210.11 MapColorScale.LeftMargin 882
2.210.12 MapColorScale.MapColorScaleTitle 882
2.210.13 MapColorScale.MapLocation 883
2.210.14 MapColorScale.MapSize 883
2.210.15 MapColorScale.NoDataText 883
2.210.16 MapColorScale.Position 884
2.210.17 MapColorScale.RangeGapColor 884
2.210.18 MapColorScale.RightMargin 885
2.210.19 MapColorScale.Style 885
2.210.20 MapColorScale.TickMarkLength 885
2.210.21 MapColorScale.ToolTip 886
2.210.22 MapColorScale.TopMargin 886
2.210.23 MapColorScale.ZIndex 886
2.211 MapColorScaleTitle 887
2.211.1 MapColorScaleTitle.Caption 887
2.211.2 MapColorScaleTitle.Style 888
2.212 MapLocation 888
2.212.1 MapLocation.Left 889
2.212.2 MapLocation.Top 889
2.212.3 MapLocation.Unit 890
2.213 MapSize 890
2.213.1 MapSize.Height 891
2.213.2 MapSize.Unit 891
2.213.3 MapSize.Width 892
2.214 MapDataRegions 892
2.214.1 MapDataRegions.MapDataRegion 893
2.215 MapDataRegion 893
2.215.1 MapDataRegion.Name 894
2.215.2 MapDataRegion.DataSetName 894
2.215.3 MapDataRegion.Filters 895
2.215.4 MapDataRegion.MapMember 895
2.215.5 MapDataRegion.Relationship 896
2.216 MapMember 896
2.216.1 MapMember.Group 897
2.216.2 MapMember.MapMember 897
2.217 MapDistanceScale 897
2.217.1 MapDistanceScale.ActionInfo 898
2.217.2 MapDistanceScale.BottomMargin 899
2.217.3 MapDistanceScale.DockOutsideViewport 899
2.217.4 MapDistanceScale.Hidden 900
2.217.5 MapDistanceScale.LeftMargin 900
2.217.6 MapDistanceScale.MapLocation 900
2.217.7 MapDistanceScale.MapSize 901
2.217.8 MapDistanceScale.Position 901

2.217.9 MapDistanceScale.RightMargin 902
2.217.10 MapDistanceScale.ScaleBorderColor 902
2.217.11 MapDistanceScale.ScaleColor 902
2.217.12 MapDistanceScale.Style 903
2.217.13 MapDistanceScale.ToolTip 903
2.217.14 MapDistanceScale.TopMargin 904
2.217.15 MapDistanceScale.ZIndex 904
2.218 MapLayers 904
2.218.1 MapLayers.MapLineLayer 905
2.218.2 MapLayers.MapPointLayer 905
2.218.3 MapLayers.MapPolygonLayer 906
2.218.4 MapLayers.MapTileLayer 906
2.219 MapLineLayer 906
2.219.1 MapLineLayer.Name 908
2.219.2 MapLineLayer.DataElementName 908
2.219.3 MapLineLayer.DataElementOutput 908
2.219.4 MapLineLayer.MapBindingFieldPairs 909
2.219.5 MapLineLayer.MapDataRegionName 909
2.219.6 MapLineLayer.MapFieldDefinitions 910
2.219.7 MapLineLayer.MapLineRules 910
2.219.8 MapLineLayer.MapLines 911
2.219.9 MapLineLayer.MapLineTemplate 911
2.219.10 MapLineLayer.MapSpatialDataRegion 911
2.219.11 MapLineLayer.MapShapefile 912
2.219.12 MapLineLayer.MapSpatialDataSet 912
2.219.13 MapLineLayer.MaximumZoom 913
2.219.14 MapLineLayer.MinimumZoom 913
2.219.15 MapLineLayer.Transparency 913
2.219.16 MapLineLayer.VisibilityMode 914
2.220 MapBindingFieldPairs 914
2.220.1 MapBindingFieldPairs.MapBindingFieldPair 915
2.221 MapBindingFieldPair 915
2.221.1 MapBindingFieldPair.BindingExpression 916
2.221.2 MapBindingFieldPair.FieldName 916
2.222 MapFieldDefinitions 917
2.222.1 MapFieldDefinitions.MapFieldDefinition 917
2.223 MapFieldDefinition 918
2.223.1 MapFieldDefinition.DataType 919
2.223.2 MapFieldDefinition.Name 919
2.224 MapLineRules 920
2.224.1 MapLineRules.MapColorPaletteRule 920
2.224.2 MapLineRules.MapColorRangeRule 921
2.224.3 MapLineRules.MapCustomColorRule 921
2.224.4 MapLineRules.MapSizeRule 922
2.225 MapColorPaletteRule 922
2.225.1 MapColorPaletteRule.BucketCount 923
2.225.2 MapColorPaletteRule.DataElementName 924
2.225.3 MapColorPaletteRule.DataElementOutput 924
2.225.4 MapColorPaletteRule.DataValue 925
2.225.5 MapColorPaletteRule.DistributionType 925
2.225.6 MapColorPaletteRule.EndValue 926
2.225.7 MapColorPaletteRule.LegendName 926
2.225.8 MapColorPaletteRule.LegendText 927
2.225.9 MapColorPaletteRule.MapBuckets 927
2.225.10 MapColorPaletteRule.Palette 927
2.225.11 MapColorPaletteRule.ShowInColorScale 928
2.225.12 MapColorPaletteRule.StartValue 928
2.226 MapBuckets 929

2.226.1 MapBuckets.MapBucket 929
2.227 MapBucket 930
2.227.1 MapBucket.EndValue 930
2.227.2 MapBucket.StartValue 931
2.228 MapColorRangeRule 931
2.228.1 MapColorRangeRule.BucketCount 932
2.228.2 MapColorRangeRule.DataElementName 933
2.228.3 MapColorRangeRule.DataElementOutput 933
2.228.4 MapColorRangeRule.DataValue 934
2.228.5 MapColorRangeRule.DistributionType 934
2.228.6 MapColorRangeRule.EndColor 935
2.228.7 MapColorRangeRule.EndValue 935
2.228.8 MapColorRangeRule.LegendName 936
2.228.9 MapColorRangeRule.LegendText 936
2.228.10 MapColorRangeRule.MapBuckets 936
2.228.11 MapColorRangeRule.MiddleColor 937
2.228.12 MapColorRangeRule.ShowInColorScale 937
2.228.13 MapColorRangeRule.StartColor 937
2.228.14 MapColorRangeRule.StartValue 938
2.229 MapCustomColorRule 938
2.229.1 MapCustomColorRule.BucketCount 939
2.229.2 MapCustomColorRule.DataElementName 940
2.229.3 MapCustomColorRule.DataElementOutput 940
2.229.4 MapCustomColorRule.DataValue 941
2.229.5 MapCustomColorRule.DistributionType 941
2.229.6 MapCustomColorRule.EndValue 942
2.229.7 MapCustomColorRule.LegendName 942
2.229.8 MapCustomColorRule.LegendText 943
2.229.9 MapCustomColorRule.MapBuckets 943
2.229.10 MapCustomColorRule.MapCustomColors 943
2.229.11 MapCustomColorRule.ShowInColorScale 944
2.229.12 MapCustomColorRule.StartValue 944
2.230 MapCustomColors 945
2.230.1 MapCustomColors.MapCustomColor 945
2.231 MapSizeRule 945
2.231.1 MapSizeRule.BucketCount 947
2.231.2 MapSizeRule.DataElementName 947
2.231.3 MapSizeRule.DataElementOutput 947
2.231.4 MapSizeRule.DataValue 948
2.231.5 MapSizeRule.DistributionType 948
2.231.6 MapSizeRule.EndSize 949
2.231.7 MapSizeRule.EndValue 949
2.231.8 MapSizeRule.LegendName 950
2.231.9 MapSizeRule.LegendText 950
2.231.10 MapSizeRule.MapBuckets 950
2.231.11 MapSizeRule.StartSize 951
2.231.12 MapSizeRule.StartValue 951
2.232 MapLines 951
2.232.1 MapLines.MapLine 952
2.233 MapLine 952
2.233.1 MapLine.MapFields 953
2.233.2 MapLine.MapLineTemplate 953
2.233.3 MapLine.UseCustomLineTemplate 954
2.233.4 MapLine.VectorData 954
2.234 MapFields 955
2.234.1 MapFields.MapField 955
2.235 MapField 955
2.235.1 MapField.Name 956

2.235.2 MapField.Value 956
2.236 MapLineTemplate 957
2.236.1 MapLineTemplate.ActionInfo 958
2.236.2 MapLineTemplate.DataElementLabel 958
2.236.3 MapLineTemplate.DataElementName 959
2.236.4 MapLineTemplate.DataElementOutput 959
2.236.5 MapLineTemplate.Hidden 959
2.236.6 MapLineTemplate.Label 960
2.236.7 MapLineTemplate.LabelPlacement 960
2.236.8 MapLineTemplate.OffsetX 961
2.236.9 MapLineTemplate.OffsetY 961
2.236.10 MapLineTemplate.Style 961
2.236.11 MapLineTemplate.ToolTip 962
2.236.12 MapLineTemplate.Width 962
2.237 MapShapefile 962
2.237.1 MapShapefile.MapFieldNames 963
2.237.2 MapShapefile.Source 963
2.238 MapFieldNames 964
2.238.1 MapFieldNames.MapFieldName 964
2.239 MapSpatialDataRegion 965
2.239.1 MapSpatialDataRegion.VectorData 965
2.240 MapSpatialDataSet 965
2.240.1 MapSpatialDataSet.DataSetName 966
2.240.2 MapSpatialDataSet.MapFieldNames 966
2.240.3 MapSpatialDataSet.SpatialField 967
2.241 MapPointLayer 967
2.241.1 MapPointLayer.Name 969
2.241.2 MapPointLayer.DataElementName 969
2.241.3 MapPointLayer.DataElementOutput 969
2.241.4 MapPointLayer.MapBindingFieldPairs 970
2.241.5 MapPointLayer.MapDataRegionName 970
2.241.6 MapPointLayer.MapFieldDefinitions 971
2.241.7 MapPointLayer.MapPointRules 971
2.241.8 MapPointLayer.MapPoints 971
2.241.9 MapPointLayer.MapMarkerTemplate 972
2.241.10 MapPointLayer.MapSpatialDataRegion 972
2.241.11 MapPointLayer.MapShapefile 973
2.241.12 MapPointLayer.MapSpatialDataSet 973
2.241.13 MapPointLayer.MaximumZoom 973
2.241.14 MapPointLayer.MinimumZoom 974
2.241.15 MapPointLayer.Transparency 974
2.241.16 MapPointLayer.VisibilityMode 975
2.242 MapMarkerTemplate 975
2.242.1 MapMarkerTemplate.ActionInfo 976
2.242.2 MapMarkerTemplate.DataElementLabel 977
2.242.3 MapMarkerTemplate.DataElementName 977
2.242.4 MapMarkerTemplate.DataElementOutput 977
2.242.5 MapMarkerTemplate.Hidden 978
2.242.6 MapMarkerTemplate.Label 978
2.242.7 MapMarkerTemplate.LabelPlacement 979
2.242.8 MapMarkerTemplate.MapMarker 979
2.242.9 MapMarkerTemplate.OffsetX 980
2.242.10 MapMarkerTemplate.OffsetY 980
2.242.11 MapMarkerTemplate.Size 980
2.242.12 MapMarkerTemplate.Style 981
2.242.13 MapMarkerTemplate.ToolTip 981
2.243 MapPointRules 982
2.243.1 MapPointRules.MapColorPaletteRule 982

2.243.2 MapPointRules.MapColorRangeRule 983
2.243.3 MapPointRules.MapCustomColorRule 983
2.243.4 MapPointRules.MapMarkerRule 984
2.243.5 MapPointRules.MapSizeRule 984
2.244 MapMarkerRule 985
2.244.1 MapMarkerRule.BucketCount 986
2.244.2 MapMarkerRule.DataElementName 986
2.244.3 MapMarkerRule.DataElementOutput 987
2.244.4 MapMarkerRule.DataValue 987
2.244.5 MapMarkerRule.DistributionType 988
2.244.6 MapMarkerRule.EndValue 988
2.244.7 MapMarkerRule.LegendName 988
2.244.8 MapMarkerRule.LegendText 989
2.244.9 MapMarkerRule.MapBuckets 989
2.244.10 MapMarkerRule.MapMarkers 990
2.244.11 MapMarkerRule.StartValue 990
2.245 MapMarkers 990
2.245.1 MapMarkers.MapMarker 991
2.246 MapMarker 991
2.246.1 MapMarker.MapMarkerImage 992
2.246.2 MapMarker.MapMarkerStyle 992
2.247 MapMarkerImage 993
2.247.1 MapMarkerImage.MIMEType 994
2.247.2 MapMarkerImage.ResizeMode 994
2.247.3 MapMarkerImage.Source 994
2.247.4 MapMarkerImage.TransparentColor 995
2.247.5 MapMarkerImage.Value 995
2.248 MapPoints 996
2.248.1 MapPoints.MapPoint 996
2.249 MapPoint 997
2.249.1 MapPoint.MapFields 997
2.249.2 MapPoint.MapMarkerTemplate 998
2.249.3 MapPoint.UseCustomPointTemplate 998
2.249.4 MapPoint.VectorData 999
2.250 MapPolygonLayer 999
2.250.1 MapPolygonLayer.Name 1001
2.250.2 MapPolygonLayer.DataElementName 1001
2.250.3 MapPolygonLayer.DataElementOutput 1001
2.250.4 MapPolygonLayer.MapBindingFieldPairs 1002
2.250.5 MapPolygonLayer.MapCenterPointRules 1002
2.250.6 MapPolygonLayer.MapMarkerTemplate 1003
2.250.7 MapPolygonLayer.MapDataRegionName 1003
2.250.8 MapPolygonLayer.MapFieldDefinitions 1003
2.250.9 MapPolygonLayer.MapPolygonRules 1004
2.250.10 MapPolygonLayer.MapPolygons 1004
2.250.11 MapPolygonLayer.MapPolygonTemplate 1005
2.250.12 MapPolygonLayer.MapSpatialDataRegion 1005
2.250.13 MapPolygonLayer.MapShapefile 1006
2.250.14 MapPolygonLayer.MapSpatialDataSet 1006
2.250.15 MapPolygonLayer.MaximumZoom 1007
2.250.16 MapPolygonLayer.MinimumZoom 1007
2.250.17 MapPolygonLayer.Transparency 1007
2.250.18 MapPolygonLayer.VisibilityMode 1008
2.251 MapPolygonRules 1008
2.251.1 MapPolygonRules.MapColorPaletteRule 1009
2.251.2 MapPolygonRules.MapColorRangeRule 1009
2.251.3 MapPolygonRules.MapCustomColorRule 1010
2.252 MapPolygons 1011

2.252.1 MapPolygons.MapPolygon 1011
2.253 MapPolygon 1011
2.253.1 MapPolygon.MapMarkerTemplate 1012
2.253.2 MapPolygon.MapFields 1013
2.253.3 MapPolygon.MapPolygonTemplate 1013
2.253.4 MapPolygon.UseCustomCenterPointTemplate 1013
2.253.5 MapPolygon.UseCustomPolygonTemplate 1014
2.253.6 MapPolygon.VectorData 1014
2.254 MapPolygonTemplate 1015
2.254.1 MapPolygonTemplate.ActionInfo 1016
2.254.2 MapPolygonTemplate.CenterPointOffsetX 1016
2.254.3 MapPolygonTemplate.CenterPointOffsetY 1017
2.254.4 MapPolygonTemplate.DataElementLabel 1017
2.254.5 MapPolygonTemplate.DataElementName 1017
2.254.6 MapPolygonTemplate.DataElementOutput 1018
2.254.7 MapPolygonTemplate.Hidden 1018
2.254.8 MapPolygonTemplate.Label 1019
2.254.9 MapPolygonTemplate.LabelPlacement 1019
2.254.10 MapPolygonTemplate.OffsetX 1020
2.254.11 MapPolygonTemplate.OffsetY 1020
2.254.12 MapPolygonTemplate.ScaleFactor 1021
2.254.13 MapPolygonTemplate.ShowLabel 1021
2.254.14 MapPolygonTemplate.Style 1021
2.254.15 MapPolygonTemplate.ToolTip 1022
2.255 MapTileLayer 1022
2.255.1 MapTileLayer.Name 1023
2.255.2 MapTileLayer.MapTiles 1023
2.255.3 MapTileLayer.MaximumZoom 1024
2.255.4 MapTileLayer.MinimumZoom 1024
2.255.5 MapTileLayer.TileStyle 1025
2.255.6 MapTileLayer.Transparency 1025
2.255.7 MapTileLayer.UseSecureConnection 1025
2.255.8 MapTileLayer.VisibilityMode 1026
2.256 MapTiles 1026
2.256.1 MapTiles.MapTile 1027
2.257 MapTile 1027
2.257.1 MapTile.Name 1028
2.257.2 MapTile.TileData 1028
2.258 MapLegends 1028
2.258.1 MapLegends.MapLegend 1029
2.259 MapLegend 1029
2.259.1 MapLegend.Name 1031
2.259.2 MapLegend.ActionInfo 1031
2.259.3 MapLegend.AutoFitTextDisabled 1032
2.259.4 MapLegend.BottomMargin 1032
2.259.5 MapLegend.DockOutsideViewport 1032
2.259.6 MapLegend.EquallySpacedItems 1033
2.259.7 MapLegend.Hidden 1033
2.259.8 MapLegend.InterlacedRows 1033
2.259.9 MapLegend.InterlacedRowsColor 1034
2.259.10 MapLegend.Layout 1034
2.259.11 MapLegend.LeftMargin 1035
2.259.12 MapLegend.MapLegendTitle 1035
2.259.13 MapLegend.MapLocation 1035
2.259.14 MapLegend.MapSize 1036
2.259.15 MapLegend.MinFontSize 1036
2.259.16 MapLegend.Position 1036
2.259.17 MapLegend.RightMargin 1037

2.259.18 MapLegend.Style 1037
2.259.19 MapLegend.TextWrapThreshold 1038
2.259.20 MapLegend.ToolTip 1038
2.259.21 MapLegend.TopMargin 1039
2.259.22 MapLegend.ZIndex 1039
2.260 MapLegendTitle 1039
2.260.1 MapLegendTitle.Caption 1040
2.260.2 MapLegendTitle.Style 1040
2.260.3 MapLegendTitle.TitleSeparator 1041
2.260.4 MapLegendTitle.TitleSeparatorColor 1041
2.261 MapTitles 1042
2.261.1 MapTitles.MapTitle 1042
2.262 MapTitle 1043
2.262.1 MapTitle.Name 1044
2.262.2 MapTitle.ActionInfo 1044
2.262.3 MapTitle.Angle 1045
2.262.4 MapTitle.BottomMargin 1045
2.262.5 MapTitle.DockOutsideViewport 1045
2.262.6 MapTitle.Hidden 1046
2.262.7 MapTitle.LeftMargin 1046
2.262.8 MapTitle.MapLocation 1046
2.262.9 MapTitle.MapSize 1047
2.262.10 MapTitle.Position 1047
2.262.11 MapTitle.RightMargin 1048
2.262.12 MapTitle.Style 1048
2.262.13 MapTitle.Text 1049
2.262.14 MapTitle.TextShadowOffset 1049
2.262.15 MapTitle.ToolTip 1049
2.262.16 MapTitle.TopMargin 1050
2.262.17 MapTitle.ZIndex 1050
2.263 MapViewport 1050
2.263.1 MapViewport.BottomMargin 1052
2.263.2 MapViewport.ContentMargin 1052
2.263.3 MapViewport.GridUnderContent 1053
2.263.4 MapViewport.LeftMargin 1053
2.263.5 MapViewport.MapCoordinateSystem 1054
2.263.6 MapViewport.MapLimits 1054
2.263.7 MapViewport.MapLocation 1054
2.263.8 MapViewport.MapMeridians 1055
2.263.9 MapViewport.MapParallels 1055
2.263.10 MapViewport.MapProjection 1055
2.263.11 MapViewport.MapSize 1056
2.263.12 MapViewport.MapCustomView 1056
2.263.13 MapViewport.MapDataBoundView 1057
2.263.14 MapViewport.MapElementView 1057
2.263.15 MapViewport.MaximumZoom 1057
2.263.16 MapViewport.MinimumZoom 1058
2.263.17 MapViewport.ProjectionCenterX 1058
2.263.18 MapViewport.ProjectionCenterY 1059
2.263.19 MapViewport.RightMargin 1059
2.263.20 MapViewport.SimplificationResolution 1059
2.263.21 MapViewport.Style 1060
2.263.22 MapViewport.TopMargin 1060
2.263.23 MapViewport.ZIndex 1060
2.264 MapCustomView 1061
2.264.1 MapCustomView.CenterX 1061
2.264.2 MapCustomView.CenterY 1062
2.264.3 MapCustomView.Zoom 1062

2.265 MapDataBoundView 1062
2.265.1 MapDataBoundView.Zoom 1063
2.266 MapElementView 1063
2.266.1 MapElementView.LayerName 1064
2.266.2 MapElementView.MapBindingFieldPairs 1064
2.266.3 MapElementView.Zoom 1065
2.267 MapLimits 1065
2.267.1 MapLimits.MaximumX 1066
2.267.2 MapLimits.MaximumY 1066
2.267.3 MapLimits.MinimumX 1066
2.267.4 MapLimits.MinimumY 1067
2.268 MapMeridians 1067
2.268.1 MapMeridians.Hidden 1068
2.268.2 MapMeridians.Interval 1068
2.268.3 MapMeridians.LabelPosition 1069
2.268.4 MapMeridians.ShowLabels 1069
2.268.5 MapMeridians.Style 1069
2.269 MapParallels 1070
2.269.1 MapParallels.Hidden 1070
2.269.2 MapParallels.Interval 1071
2.269.3 MapParallels.LabelPosition 1071
2.269.4 MapParallels.ShowLabels 1072
2.269.5 MapParallels.Style 1072
2.270 CustomData 1072
2.270.1 CustomData.DataColumnHierarchy 1074
2.270.2 CustomData.DataRowHierarchy 1075
2.270.3 CustomData.DataRows 1075
2.270.4 CustomData.DataSetName 1075
2.270.5 CustomData.Filters 1076
2.270.6 CustomData.SortExpressions 1076
2.270.7 CustomData.DataColumnGroupings 1076
2.270.8 CustomData.DataRowGroupings 1077
2.270.9 CustomData.Relationship 1077
2.271 DataColumnHierarchy 1078
2.271.1 DataColumnHierarchy.DataMembers 1078
2.272 DataMembers 1079
2.272.1 DataMembers.DataMember 1079
2.273 DataMember 1080
2.273.1 DataMember.CustomProperties 1081
2.273.2 DataMember.DataMembers 1081
2.273.3 DataMember.Group 1081
2.273.4 DataMember.SortExpressions 1082
2.273.5 DataMember.Subtotal 1082
2.274 DataRowHierarchy 1082
2.274.1 DataRowHierarchy.DataMembers 1083
2.275 DataRows 1083
2.275.1 DataRows.DataRow 1084
2.276 DataRow 1084
2.276.1 DataRow.DataCell 1085
2.277 DataCell 1085
2.277.1 DataCell.DataValue 1086
2.277.2 DataCell.DataSetName 1087
2.277.3 DataCell.Relationships 1087
2.278 DataValue 1088
2.278.1 DataValue.Name 1088
2.278.2 DataValue.Value 1088
2.279 DataColumnGroupings 1089
2.279.1 DataColumnGroupings.DataGroupings 1089

2.280 DataGroupings 1090
2.280.1 DataGroupings.DataGrouping 1090
2.281 DataGrouping 1090
2.281.1 DataGrouping.CustomProperties 1091
2.281.2 DataGrouping.DataGroupings 1091
2.281.3 DataGrouping.Grouping 1092
2.281.4 DataGrouping.Sorting 1092
2.281.5 DataGrouping.Static 1093
2.281.6 DataGrouping.Subtotal 1093
2.282 DataRowGroupings 1093
2.282.1 DataRowGroupings.DataGroupings 1094
2.283 CustomReportItem 1094
2.283.1 CustomReportItem.Name 1098
2.283.2 CustomReportItem.Type 1099
2.283.3 CustomReportItem.ActionInfo 1099
2.283.4 CustomReportItem.AltReportItem 1099
2.283.5 CustomReportItem.Bookmark 1100
2.283.6 CustomReportItem.CustomData 1100
2.283.7 CustomReportItem.CustomProperties 1101
2.283.8 CustomReportItem.DataElementName 1101
2.283.9 CustomReportItem.DataElementOutput 1101
2.283.10 CustomReportItem.DocumentMapLabel 1102
2.283.11 CustomReportItem.Height 1102
2.283.12 CustomReportItem.Label 1103
2.283.13 CustomReportItem.Left 1103
2.283.14 CustomReportItem.RepeatWith 1103
2.283.15 CustomReportItem.ReportItems 1104
2.283.16 CustomReportItem.Style 1104
2.283.17 CustomReportItem.ToolTip 1104
2.283.18 CustomReportItem.Top 1105
2.283.19 CustomReportItem.Visibility 1105
2.283.20 CustomReportItem.Width 1106
2.283.21 CustomReportItem.ZIndex 1106
2.284 ActionInfo 1106
2.284.1 ActionInfo.Actions 1108
2.285 Actions 1109
2.285.1 Actions.Action 1109
2.286 Action 1110
2.286.1 Action.BookmarkLink 1111
2.286.2 Action.Drillthrough 1112
2.286.3 Action.Hyperlink 1112
2.286.4 Action.Label 1112
2.287 Drillthrough 1113
2.287.1 Drillthrough.Parameters 1113
2.287.2 Drillthrough.ReportName 1114
2.288 Custom 1114
2.289 CustomProperties 1115
2.289.1 CustomProperties.CustomProperty 1116
2.290 CustomProperty 1116
2.290.1 CustomProperty.Name 1117
2.290.2 CustomProperty.Value 1118
2.291 PageBreak 1118
2.291.1 PageBreak.BreakLocation 1119
2.291.2 PageBreak.Disabled 1120
2.291.3 PageBreak.ResetPageNumber 1120
2.292 Style 1120
2.292.1 Style.BackgroundColor 1126
2.292.2 Style.BackgroundGradientEndColor 1127

2.292.3 Style.BackgroundGradientType 1127
2.292.4 Style.BackgroundHatchType 1128
2.292.5 Style.BackgroundImage 1131
2.292.6 Style.Border 1131
2.292.7 Style.BorderColor 1131
2.292.8 Style.BorderStyle 1131
2.292.9 Style.BorderWidth 1132
2.292.10 Style.BottomBorder 1132
2.292.11 Style.Calendar 1133
2.292.12 Style.Color 1133
2.292.13 Style.CurrencyLanguage 1134
2.292.14 Style.Direction 1134
2.292.15 Style.FontFamily 1135
2.292.16 Style.FontSize 1136
2.292.17 Style.FontStyle 1136
2.292.18 Style.FontWeight 1136
2.292.19 Style.Format 1137
2.292.20 Style.Language 1138
2.292.21 Style.LeftBorder 1138
2.292.22 Style.LineHeight 1139
2.292.23 Style.NumeralLanguage 1139
2.292.24 Style.NumeralVariant 1140
2.292.25 Style.PaddingBottom 1140
2.292.26 Style.PaddingLeft 1141
2.292.27 Style.PaddingRight 1141
2.292.28 Style.PaddingTop 1142
2.292.29 Style.RightBorder 1142
2.292.30 Style.ShadowColor 1142
2.292.31 Style.ShadowOffset 1143
2.292.32 Style.TextAlign 1143
2.292.33 Style.TextDecoration 1144
2.292.34 Style.TextEffect 1144
2.292.35 Style.TopBorder 1145
2.292.36 Style.UnicodeBiDi 1145
2.292.37 Style.VerticalAlign 1146
2.292.38 Style.WritingMode 1146
2.293 BackgroundImage 1147
2.293.1 BackgroundImage.BackgroundRepeat 1148
2.293.2 BackgroundImage.MIMEType 1149
2.293.3 BackgroundImage.Position 1149
2.293.4 BackgroundImage.Source 1150
2.293.5 BackgroundImage.TransparentColor 1151
2.293.6 BackgroundImage.Value 1151
2.293.7 BackgroundImage.EmbeddingMode 1152
2.293.8 BackgroundImage.Transparency 1152
2.294 Border 1153
2.294.1 Border.Color 1154
2.294.2 Border.Style 1154
2.294.3 Border.Width 1155
2.295 BorderColor 1155
2.295.1 BorderColor.Bottom 1156
2.295.2 BorderColor.Default 1156
2.295.3 BorderColor.Left 1156
2.295.4 BorderColor.Right 1157
2.295.5 BorderColor.Top 1157
2.296 BorderStyle 1157
2.296.1 BorderStyle.Bottom 1158
2.296.2 BorderStyle.Default 1159

2.296.3 BorderStyle.Left 1159
2.296.4 BorderStyle.Right 1160
2.296.5 BorderStyle.Top 1161
2.297 BorderWidth 1162
2.297.1 BorderWidth.Bottom 1162
2.297.2 BorderWidth.Default 1163
2.297.3 BorderWidth.Left 1163
2.297.4 BorderWidth.Right 1163
2.297.5 BorderWidth.Top 1164
2.298 Parameters 1164
2.298.1 Parameters.Parameter 1165
2.299 Parameter 1165
2.299.1 Parameter.Name 1165
2.299.2 Parameter.Omit 1166
2.299.3 Parameter.Value 1166
2.300 Visibility 1166
2.300.1 Visibility.Hidden 1168
2.300.2 Visibility.ToggleItem 1168
2.301 Classes 1168
2.301.1 Classes.Class 1169
2.302 Class 1169
2.302.1 Class.ClassName 1170
2.302.2 Class.InstanceName 1170
2.303 CodeModules 1170
2.303.1 CodeModules.CodeModule 1171
2.304 EmbeddedImages 1171
2.304.1 EmbeddedImages.EmbeddedImage 1172
2.305 EmbeddedImage 1172
2.305.1 EmbeddedImage.Name 1173
2.305.2 EmbeddedImage.ImageData 1173
2.305.3 EmbeddedImage.MIMEType 1173
2.306 ReportParameters 1174
2.306.1 ReportParameters.ReportParameter 1174
2.307 ReportParameter 1174
2.307.1 ReportParameter.Name 1176
2.307.2 ReportParameter.AllowBlank 1176
2.307.3 ReportParameter.DataType 1176
2.307.4 ReportParameter.DefaultValue 1177
2.307.5 ReportParameter.Hidden 1177
2.307.6 ReportParameter.MultiValue 1178
2.307.7 ReportParameter.Nullable 1178
2.307.8 ReportParameter.Prompt 1179
2.307.9 ReportParameter.UsedInQuery 1179
2.307.10 ReportParameter.ValidValues 1180
2.308 ReportParametersLayout 1180
2.308.1 ReportParametersLayout.GridLayoutDefinition 1180
2.309 GridLayoutDefinition 1181
2.309.1 GridLayoutDefinition.NumberOfColumns 1181
2.309.2 GridLayoutDefinition.NumberOfRows 1182
2.309.3 GridLayoutDefinition.CellDefinitions 1182
2.310 CellDefinitions 1182
2.310.1 CellDefinitions.CellDefinition 1183
2.311 CellDefinition 1183
2.311.1 CellDefinition.ColumnIndex 1184
2.311.2 CellDefinition.RowIndex 1184
2.311.3 CellDefinition.ParameterName 1185
2.312 DefaultValue 1185
2.312.1 DefaultValue.DataSetReference 1186

2.312.2 DefaultValue.Values 1186
2.313 DataSetReference 1186
2.313.1 DataSetReference.DataSetName 1187
2.313.2 DataSetReference.LabelField 1187
2.313.3 DataSetReference.ValueField 1188
2.314 Values 1188
2.314.1 Values.Value 1189
2.315 ValidValues 1189
2.315.1 ValidValues.DataSetReference 1190
2.315.2 ValidValues.ParameterValues 1190
2.316 ParameterValues 1190
2.316.1 ParameterValues.ParameterValue 1191
2.317 ParameterValue 1191
2.317.1 ParameterValue.Label 1192
2.317.2 ParameterValue.Value 1192
2.318 Variables 1193
2.318.1 Variables.Variable 1193
2.319 Variable 1194
2.319.1 Variable.Name 1195
2.319.2 Variable.Value 1195
2.319.3 Variable.Value.DataType 1195
2.319.4 Variable.Writable 1196
2.320 DataSources 1196
2.320.1 DataSources.DataSource 1197
2.321 DataSource 1197
2.321.1 DataSource.Name 1198
2.321.2 DataSource.ConnectionProperties 1198
2.321.3 DataSource.DataSourceReference 1198
2.321.4 DataSource.Transaction 1199
2.322 ConnectionProperties 1199
2.322.1 ConnectionProperties.ConnectString 1200
2.322.2 ConnectionProperties.DataProvider 1200
2.322.3 ConnectionProperties.IntegratedSecurity 1200
2.322.4 ConnectionProperties.Prompt 1200
2.323 DataSets 1201
2.323.1 DataSets.DataSet 1201
2.324 DataSet 1202
2.324.1 DataSet.Name 1205
2.324.2 DataSet.AccentSensitivity 1205
2.324.3 DataSet.CaseSensitivity 1206
2.324.4 DataSet.Collation 1206
2.324.5 DataSet.Fields 1212
2.324.6 DataSet.Filters 1212
2.324.7 DataSet.InterpretSubtotalsAsDetails 1212
2.324.8 DataSet.KanatypeSensitivity 1213
2.324.9 DataSet.Query 1213
2.324.10 DataSet.SharedDataSet 1214
2.324.11 DataSet.WidthSensitivity 1214
2.324.12 DataSet.DefaultRelationships 1215
2.324.13 DataSet.NullsAsBlanks 1215
2.324.14 DataSet.CollationCulture 1215
2.325 SharedDataSet 1216
2.325.1 SharedDataSet.SharedDataSetReference 1216
2.325.2 SharedDataSet.QueryParameters 1217
2.326 Fields 1217
2.326.1 Fields.Field 1217
2.327 Field 1218
2.327.1 Field.Name 1219

2.327.2 Field.DataField 1219
2.327.3 Field.Value 1219
2.327.4 Field.Value.DataType 1220
2.327.5 Field.AggregateIndicatorField 1220
2.328 Filters 1221
2.328.1 Filters.Filter 1221
2.329 Filter 1222
2.329.1 Filter.FilterExpression 1223
2.329.2 Filter.FilterValues 1223
2.329.3 Filter.Operator 1223
2.330 FilterValues 1225
2.330.1 FilterValues.FilterValue 1226
2.330.2 FilterValues.FilterValue.DataType 1227
2.331 Query 1227
2.331.1 Query.CommandText 1228
2.331.2 Query.CommandType 1228
2.331.3 Query.DataSourceName 1229
2.331.4 Query.QueryParameters 1229
2.331.5 Query.Timeout 1229
2.332 QueryParameters 1230
2.332.1 QueryParameters.QueryParameter 1230
2.333 QueryParameter 1231
2.333.1 QueryParameter.Name 1232
2.333.2 QueryParameter.Value 1232
2.333.3 QueryParameter.Value.DataType 1233
2.334 DefaultRelationships 1233
2.334.1 DefaultRelationships.DefaultRelationship 1233
2.335 DefaultRelationship 1234
2.335.1 DefaultRelationship.RelatedDataSet 1234
2.335.2 DefaultRelationship.JoinConditions 1235
2.335.3 DefaultRelationship.NaturalJoin 1235
2.336 JoinConditions 1235
2.336.1 JoinConditions.JoinCondition 1236
2.337 JoinCondition 1236
2.337.1 JoinCondition.ForeignKey 1237
2.337.2 JoinCondition.PrimaryKey 1237
2.337.3 JoinCondition.SortDirection 1238
2.338 Relationships 1238
2.338.1 Relationships.Relationship 1239
2.339 Relationship 1239
2.339.1 Relationship.ParentScope 1240
2.339.2 Relationship.NaturalJoin 1241
2.339.3 Relationship.JoinConditions 1241
2.340 Expressions 1241
2.340.1 Expression Syntax 1241
2.340.2 Custom Code References 1242
2.340.3 Built-in Code Namespace References 1242
2.340.4 Data Types 1242
2.340.5 Expression Error Handling 1242
2.340.6 Global Collections 1243
2.340.6.1 Fields 1243
2.340.6.2 Predefined Field Properties 1244
2.340.6.3 Fields Collection in Reports with Multiple Datasets 1245
2.340.6.4 Parameters 1245
2.340.6.5 ReportItems 1245
2.340.6.6 Globals 1246
2.340.6.7 ReadOnlyNameValueCollection 1246
2.340.6.8 User 1247

2.340.6.9 DataSources 1247
2.340.6.10 DataSets 1247
2.340.6.11 Variables 1248
2.340.6.12 Restrictions on Use of Global Collections 1248
2.340.6.13 Scopes 1250
2.340.7 Aggregate Functions 1250
2.340.7.1 Scope 1251
2.340.7.2 Recursive 1251
2.340.7.3 Recursive Depth 1252
2.340.7.4 Function: Level 1252
2.340.7.5 Aggregate Function: Sum 1252
2.340.7.6 Aggregate Function: Avg 1253
2.340.7.7 Aggregate Function: Max 1253
2.340.7.8 Aggregate Function: Min 1254
2.340.7.9 Aggregate Function: Count 1254
2.340.7.10 Aggregate Function: CountDistinct 1254
2.340.7.11 Aggregate Function: CountRows 1255
2.340.7.12 Aggregate Function: StDev 1255
2.340.7.13 Aggregate Function: StDevP 1255
2.340.7.14 Aggregate Function: Var 1256
2.340.7.15 Aggregate Function: VarP 1256
2.340.7.16 Aggregate Function: Union 1256
2.340.7.17 Aggregate Function: First 1256
2.340.7.18 Aggregate Function: Last 1257
2.340.7.19 Aggregate Function: Previous 1257
2.340.7.20 Aggregate Function: RunningValue 1258
2.340.7.21 Aggregate Function: RowNumber 1258
2.340.7.22 Aggregate Function: Aggregate 1259
2.340.7.23 Restrictions on Aggregate Usage 1259
2.340.7.24 Restrictions on Aggregate Nesting 1260
2.340.7.25 Filtering and Aggregates 1260
2.340.7.26 Additional Functions 1261
2.340.7.26.1 Dynamic Scoping 1261
2.340.7.26.2 Function: InScope 1261
2.340.7.26.3 Semantic Query Drillthrough 1261
2.340.7.26.4 Function: CreateDrillthroughContext 1262
2.340.7.26.5 Function: Lookup 1262
2.340.7.26.6 Function: LookupSet 1262
2.340.7.26.7 Function: MultiLookup 1263
2.340.7.26.8 Function: MinValue 1264
2.340.7.26.9 Function: MaxValue 1264

3 Structure Examples 1266
3.1 Report in RDL schema 2008/01 1266
3.2 Report in RDL schema 2010/01 1270
3.3 Report in RDL schema 2005/01 1276
3.4 Column and Line Chart 1279
3.5 Doughnut Chart 1297
3.6 CustomReportItem 1300
3.7 Tablix 1 1305
3.8 Tablix 2 1307
3.9 Tablix 3 1313
3.10 Data 1326
3.11 Subreport 1327
3.12 TextBox 1327
3.13 GaugePanel 1330
3.14 Line 1336
3.15 List 1337

3.16 Rectangle with Image and Textbox 1341
3.17 Map 1343
3.18 Matrix 1352
3.19 Table 1358

4 Security 1372

5 Appendix A: RDL XML Schemas 1373
5.1 RDL XML Schema for Version 2003/10 1373
5.2 RDL XML Schema for Version 2005/01 1395
5.3 RDL XML Schema for Version 2008/01 1421
5.4 RDL XML Schema for Version 2010/01 1463
5.5 RDL XML Schema for Version 2011/01 1519
5.6 RDL XML Schema for Version 2012/01 1523
5.7 RDL XML Schema for Version 2013/01 1524
5.8 RDL XML Schema for Version 2016/01 1526

6 Appendix B: Error Codes 1582
6.1 rsAggregateInFilterExpression 1582
6.2 rsAggregateInGroupExpression 1582
6.3 rsAggregateInQueryParameterExpression 1582
6.4 rsAggregateInReportParameterExpression 1582
6.5 rsAggregateInReportLanguageExpression 1582
6.6 rsAggregateInCalculatedFieldExpression 1582
6.7 rsAggregateofAggregate 1582
6.8 rsAggregateReportItemInBody 1582
6.9 rsBinaryConstant 1582
6.10 rsChartSeriesPlotTypeIgnored 1583
6.11 rsCompilerErrorInExpression 1583
6.12 rsCompilerErrorInCode 1583
6.13 rsCompilerErrorInClassInstanceDeclaration 1583
6.14 rsUnexpectedCompilerError 1583
6.15 rsConflictingRunningValueScopesInMatrix 1583
6.16 rsConflictingRunningValueScopesInTablix 1583
6.17 rsCountRowsInPageSectionExpression 1584
6.18 rsCountStarNotSupported 1584
6.19 rsCountStarRVNotSupported 1584
6.20 rsCustomAggregateAndFilter 1584
6.21 rsDataRegionInDetailList 1584
6.22 rsDataRegionInPageSection 1584
6.23 rsDataRegionInTableDetailRow 1584
6.24 rsDataRegionWithoutDataSet 1584
6.25 rsDataSourceReferenceNotPublished 1585
6.26 rsDuplicateChartAxisName 1585
6.27 rsSpecifiedNonValueAxisName 1585
6.28 rsValueAxisNameNotFound 1585
6.29 rsInvalidTextEffect 1585
6.30 rsInvalidBackgroundHatchType 1585
6.31 rsInvalidBackgroundImagePosition 1585
6.32 rsPageBreakOnGaugeGroup 1585
6.33 rsDuplicateChartLegendCustomItemCellName 1585
6.34 rsDuplicateChartFormulaParameter 1586
6.35 rsDuplicateClassInstanceName 1586
6.36 rsDuplicateDataSourceName 1586
6.37 rsInvalidDataSourceNameLength 1586
6.38 rsDuplicateEmbeddedImageName 1586
6.39 rsInvalidEmbeddedImageNameNotCLSCompliant 1586
6.40 rsInvalidEmbeddedImageNameLength 1586
6.41 rsDuplicateFieldName 1586

6.42 rsDuplicateReportItemName 1586
6.43 rsDuplicateReportParameterName 1587
6.44 rsDuplicateCaseInsensitiveReportParameterName 1587
6.45 rsDuplicateScopeName 1587
6.46 rsExpressionMissingCloseParen 1587
6.47 rsFieldInPageSectionExpression 1587
6.48 rsFieldInReportParameterExpression 1587
6.49 rsFieldInQueryParameterExpression 1587
6.50 rsFieldInReportLanguageExpression 1587
6.51 rsGlobalNotDefined 1587
6.52 rsInvalidAction 1587
6.53 rsInvalidAggregateScope 1588
6.54 rsInvalidAltReportItem 1588
6.55 rsInvalidBooleanConstant 1588
6.56 rsInvalidCategoryGrouping 1588
6.57 rsInvalidCharacterInExpression 1588
6.58 rsInvalidChartGroupings 1588
6.59 rsInvalidChartSubType 1588
6.60 rsInvalidColumnGrouping 1588
6.61 rsInvalidCustomAggregateExpression 1589
6.62 rsInvalidCustomAggregateScope 1589
6.63 rsInvalidCustomPropertyName 1589
6.64 rsInvalidDataElementNameNotCLSCompliant 1589
6.65 rsInvalidDataSetName 1589
6.66 rsInvalidDataSource 1589
6.67 rsInvalidDataSourceReference 1589
6.68 rsInvalidValidValuesDataSetReference 1589
6.69 rsInvalidDefaultValueDataSetReference 1589
6.70 rsInvalidDataSetReferenceField 1590
6.71 rsInvalidDefaultValue 1590
6.72 rsInvalidDefaultValueValues 1590
6.73 rsInvalidDetailsDataGrouping 1590
6.74 rsInvalidExpressionScope 1590
6.75 rsInvalidExpressionScopeDataSet 1590
6.76 rsInvalidSortExpressionScope 1590
6.77 rsIneffectiveSortExpressionScope 1590
6.78 rsInvalidField 1591
6.79 rsInvalidFieldNameNotCLSCompliant 1591
6.80 rsInvalidFieldNameLength 1591
6.81 rsInvalidGroupExpressionScope 1591
6.82 rsInvalidGroupNameNotCLSCompliant 1591
6.83 rsInvalidHideDuplicateScope 1591
6.84 rsInvalidURLProtocol 1591
6.85 rsInvalidIntegerConstant 1591
6.86 rsInvalidDateTimeConstant 1591
6.87 rsInvalidFloatConstant 1592
6.88 rsLabelExpressionOnChartScalarAxisIsIgnored 1592
6.89 rsInvalidMatrixSubtotalReportItem 1592
6.90 rsInvalidNameNotCLSCompliant 1592
6.91 rsInvalidNameLength 1592
6.92 rsInvalidNumberOfFilterValues 1592
6.93 rsInvalidOmittedExpressionScope 1592
6.94 rsInvalidOmittedTargetScope 1592
6.95 rsInvalidParameterNameNotCLSCompliant 1592
6.96 rsInvalidParameterNameLength 1593
6.97 rsInvalidPreviousAggregateInMatrixCell 1593
6.98 rsInvalidPreviousAggregateInTablixCell 1593
6.99 rsInvalidRepeatWith 1593

6.100 rsInvalidReportDefinition 1593
6.101 rsInvalidReportParameterDependency 1593
6.102 rsInvalidRowGrouping 1593
6.103 rsInvalidRunningValueAggregate 1593
6.104 rsInvalidScopeInMatrix 1593
6.105 rsInvalidScopeInTablix 1594
6.106 rsInvalidSeriesGrouping 1594
6.107 rsInvalidStaticDataGrouping 1594
6.108 rsInvalidReportName 1594
6.109 rsInvalidReportNameCharacters 1594
6.110 rsInvalidReportUri 1595
6.111 rsInvalidTargetScope 1595
6.112 rsInvalidTextboxInPageSection 1595
6.113 rsInvalidReportItemInPageSection 1595
6.114 rsInvalidToggleItem 1595
6.115 rsInvalidValidValues 1595
6.116 rsInvalidMultiValueParameter 1595
6.117 rsMapLayerMissingProperty 1595
6.118 rsMapPropertyAlreadyDefined 1596
6.119 rsMissingAggregateScope 1596
6.120 rsMissingChartDataPoints 1596
6.121 rsMissingDataSetName 1596
6.122 rsMissingMIMEType 1596
6.123 rsMissingParameterDefault 1596
6.124 rsMultipleGroupExpressionsOnChartScalarAxis 1596
6.125 rsMultipleGroupingsOnChartScalarAxis 1596
6.126 rsMultiReportItemsInMatrixSection 1597
6.127 rsMultiReportItemsInTablixCell 1597
6.128 rsMultiReportItemsInPageSectionExpression 1597
6.129 rsMultiReportItemsInCustomReportItem 1597
6.130 rsMultiStaticCategoriesOrSeries 1597
6.131 rsMultiStaticColumnsOrRows 1597
6.132 rsNegativeLeftWidth 1597
6.133 rsNegativeTopHeight 1597
6.134 rsNonAggregateInMatrixCell 1598
6.135 rsNonExistingScope 1598
6.136 rsNotAReportDefinition 1598
6.137 rsNotACurrentReportDefinition 1598
6.138 rsOverlappingReportItems 1598
6.139 rsReportItemOutsideContainer 1598
6.140 rsPageBreakOnMatrixColumnGroup 1598
6.141 rsPageBreakOnChartGroup 1598
6.142 rsParameterValueNullOrBlank 1599
6.143 rsPreviousAggregateInFilterExpression 1599
6.144 rsPreviousAggregateInGroupExpression 1599
6.145 rsPreviousAggregateInPageSectionExpression 1599
6.146 rsPreviousAggregateInQueryParameterExpression 1599
6.147 rsPreviousAggregateInReportParameterExpression 1599
6.148 rsPreviousAggregateInReportLanguageExpression 1599
6.149 rsPreviousAggregateInSortExpression 1599
6.150 rsRepeatWithNotPeerDataRegion 1599
6.151 rsReportItemInFilterExpression 1599
6.152 rsReportItemInGroupExpression 1600
6.153 rsReportItemInQueryParameterExpression 1600
6.154 rsReportItemInReportParameterExpression 1600
6.155 rsReportItemInSortExpression 1600
6.156 rsReportItemInReportLanguageExpression 1600
6.157 rsReportItemInVariableExpression 1600

6.158 rsReportParameterPropertyTypeMismatch 1600
6.159 rsRowNumberInFilterExpression 1600
6.160 rsRowNumberInPageSectionExpression 1600
6.161 rsRowNumberInQueryParameterExpression 1600
6.162 rsRowNumberInReportParameterExpression 1601
6.163 rsRowNumberInReportLanguageExpression 1601
6.164 rsRowNumberInSortExpression 1601
6.165 rsRowNumberInVariableExpression 1601
6.166 rsRunningValueInFilterExpression 1601
6.167 rsRunningValueInGroupExpression 1601
6.168 rsRunningValueInPageSectionExpression 1601
6.169 rsRunningValueInQueryParameterExpression 1601
6.170 rsRunningValueInReportParameterExpression 1601
6.171 rsRunningValueInReportLanguageExpression 1601
6.172 rsRunningValueInSortExpression 1601
6.173 rsRunningValueInVariableExpression 1602
6.174 rsScopeInPageSectionExpression 1602
6.175 rsStaticGroupingOnChartScalarAxis 1602
6.176 rsToggleInPageSection 1602
6.177 rsUnsortedCategoryInAreaChart 1602
6.178 rsWrongNumberOfMatrixCells 1602
6.179 rsWrongNumberOfMatrixColumns 1602
6.180 rsWrongNumberOfMatrixRows 1602
6.181 rsWrongNumberOfChartDataPoints 1603
6.182 rsWrongNumberOfChartSeries 1603
6.183 rsWrongNumberOfChartDataPointsInSeries 1603
6.184 rsWrongNumberOfDataValues 1603
6.185 rsWrongNumberOfParameters 1603
6.186 rsWrongNumberOfTableCells 1603
6.187 rsMissingDataGrouping 1603
6.188 rsWrongNumberOfDataRows 1604
6.189 rwWrongNumberOfDataCellsInDataRow 1604
6.190 rsInvalidRecursiveAggregate 1604
6.191 rsInvalidAggregateRecursiveFlag 1604
6.192 rsPostSortAggregateInGroupFilterExpression 1604
6.193 rsPostSortAggregateInSortExpression 1604
6.194 rsPostSortAggregateInVariableExpression 1604
6.195 rsAggregateInPreviousAggregate 1604
6.196 rsRunningValueInPreviousAggregate 1604
6.197 rsPreviousInPreviousAggregate 1605
6.198 rsRowNumberInPreviousAggregate 1605
6.199 rsInScopeOrLevelInPreviousAggregate 1605
6.200 rsInvalidScopeInInnerAggregateOfPreviousAggregate 1605
6.201 rsInvalidGroupingParent 1605
6.202 rsMissingDataGroupings 1605
6.203 rsMissingDataCells 1605
6.204 rsCRIMultiStaticColumnsOrRows 1605
6.205 rsCRIStaticWithSubgroups 1605
6.206 rsCRIMultiNonStaticGroups 1606
6.207 rsCRISubtotalNotSupported 1606
6.208 rsInvalidGrouping 1606
6.209 rsCRIInPageSection 1606
6.210 rsBookmarkInPageSection 1606
6.211 rsCantMakeTableGroupHeadersFixed 1606
6.212 rsFixedHeadersInInnerDataRegion 1606
6.213 rsInvalidFixedTableColumnHeaderSpacing 1606
6.214 rsUnsupportedProtocol 1607
6.215 rsVariableInPreviousAggregate 1607

6.216 rsAggregateOfVariable 1607
6.217 rsVariableInQueryParameterExpression 1607
6.218 rsVariableInReportParameterExpression 1607
6.219 rsVariableInReportLanguageExpression 1607
6.220 rsVariableInGroupExpression 1607
6.221 rsVariableInCalculatedFieldExpression 1607
6.222 rsDataSetInPageSectionExpression 1607
6.223 rsDataSetInQueryParameterExpression 1608
6.224 rsDataSetInReportParameterExpression 1608
6.225 rsDataSetInReportLanguageExpression 1608
6.226 rsDataSourceInPageSectionExpression 1608
6.227 rsDataSourceInQueryParameterExpression 1608
6.228 rsDataSourceInReportParameterExpression 1608
6.229 rsDataSourceInReportLanguageExpression 1608
6.230 rsInvalidMeDotValueInExpression 1608
6.231 rsWrongNumberOfTablixCornerRows 1608
6.232 rsWrongNumberOfTablixCornerCells 1608
6.233 rsWrongNumberOfTablixColumns 1609
6.234 rsWrongNumberOfTablixCells 1609
6.235 rsWrongNumberOfTablixRows 1609
6.236 rsInvalidTablixCornerCellSpan 1609
6.237 rsInvalidTablixCornerRowSpans 1609
6.238 rsInvalidTablixCornerColumnSpans 1609
6.239 rsInvalidSortNotAllowed 1610
6.240 rsInvalidFixedHeaderOnOppositeHierarchy 1610
6.241 rsInvalidFixedDataColumnPosition 1610
6.242 rsInvalidFixedDataRowPosition 1610
6.243 rsInvalidFixedDataNotContiguous 1610
6.244 rsInvalidFixedDataInHierarchy 1610
6.245 rsInvalidKeepWithGroup 1610
6.246 rsInvalidKeepWithGroupOnDynamicTablixMember 1611
6.247 rsInvalidKeepWithGroupOnColumnTablixMember 1611
6.248 rsInvalidRepeatOnNewPageOnColumnTablixMember 1611
6.249 rsInvalidRepeatOnNewPage 1611
6.250 rsInvalidTablixCellColSpans 1611
6.251 rsInvalidTablixCellColSpan 1611
6.252 rsInvalidTablixCellRowSpan 1611
6.253 rsCellContentsNotOmitted 1612
6.254 rsCellContentsRequired 1612
6.255 rsInvalidTablixCellCellSpan 1612
6.256 rsInconsistentNumberOfCellsInRow 1612
6.257 rsInvalidTablixHeaderSize 1612
6.258 rsInvalidTablixHeaders 1612
6.259 rsInvalidInnerDataSetName 1612
6.260 rsDuplicateVariableName 1613
6.261 rsInvalidVariableReference 1613
6.262 rsInvalidVariableNameNotCLSCompliant 1613
6.263 rsInvalidVariableNameLength 1613
6.264 rsInvalidGroupingVariableNameNotCLSCompliant 1613
6.265 rsInvalidGroupingVariableNameLength 1613
6.266 rsInvalidVariableCount 1613
6.267 rsMissingExpression 1613
6.268 rsInvalidActionsCount 1614
6.269 rsInvalidFixedDataBodyCellSpans 1614
6.270 rsInvalidEmptyImageReference 1614
6.271 rsFieldReference 1614
6.272 rsInvalidBackgroundRepeat 1614
6.273 rsInvalidBackgroundGradientType 1614

6.274 rsInvalidBorderStyle 1614
6.275 rsInvalidCalender 1614
6.276 rsInvalidCalendarForLanguage 1615
6.277 rsInvalidColor 1615
6.278 rsInvalidDirection 1615
6.279 rsInvalidEmbeddedImageProperty 1615
6.280 rsInvalidFontStyle 1615
6.281 rsInvalidFontWeight 1615
6.282 rsInvalidLanguage 1615
6.283 rsInvalidMeasurementUnit 1615
6.284 rsInvalidMIMEType 1615
6.285 rsInvalidNumeralVariant 1615
6.286 rsInvalidNumeralVariantForLanguage 1616
6.287 rsInvalidSize 1616
6.288 rsInvalidTextAlign 1616
6.289 rsInvalidTextDecoration 1616
6.290 rsInvalidUnicodeBiDi 1616
6.291 rsInvalidVerticalAlign 1616
6.292 rsInvalidWritingMode 1616
6.293 rsNegativeSize 1616
6.294 rsOutOfRangeSize 1616
6.295 rsPageNumberInBody 1617
6.296 rsParameterReference 1617
6.297 rsReportItemReference 1617
6.298 rsDataSetReference 1617
6.299 rsDataSourceReference 1617
6.300 rsErrorLoadingCodeModule 1617
6.301 rsInvalidObjectNameNotUnique 1617
6.302 rsInvalidObjectNameNotCLSCompliant 1617
6.303 rsInvalidSourceSeriesName 1617
6.304 rsInvalidDataSourceNameNotCLSCompliant 1618
6.305 rsDuplicateChartLegendItemName 1618
6.306 rsInvalidEnumValue 1618
6.307 rsInvalidListStyle 1618
6.308 rsInvalidMarkupType 1618
6.309 rsMissingAggregateScopeInPageSection 1618
6.310 rsReportItemInScopedAggregate 1618
6.311 rsPageNumberInScopedAggregates 1618
6.312 rsVariableInDataRowSortExpression 1619
6.313 rsAggregateInDataRowSortExpression 1619
6.314 rsVariableInDataRegionOrDataSetFilterExpression 1619
6.315 rsNestedLookups 1619
6.316 rsLookupInFilterExpression 1619
6.317 rsInvalidLookupScope 1619
6.318 rsLookupOfVariable 1619
6.319 rsReportItemInLookupDestinationOrResult 1620
6.320 rsAggregateInLookupDestinationOrResult 1620
6.321 rsPagePropertyInSubsequentReportSection 1620
6.322 rsReportItemReferenceInPageSection 1620
6.323 rsInvalidColumnsInReportSection 1620
6.324 rsRowNumberInLookupDestinationOrResult 1620
6.325 rsPreviousInLookupDestinationOrResult 1620
6.326 rsLevelCallRecursiveHierarchyBothDimensions 1621
6.327 rsInvalidColumnsInBody 1621
6.328 rsDuplicateGroupingVariableName 1621
6.329 rsNestedCustomAggregate 1621
6.330 rsInvalidNestedAggregateScope 1621
6.331 rsNestedAggregateScopesFromDifferentAxes 1621

6.332 rsIncompatibleNestedAggregateScopes 1622
6.333 rsNestedAggregateScopeRequired 1622
6.334 rsInvalidNestedDataSetAggregate 1622
6.335 rsDataSetAggregateOfAggregates 1622
6.336 rsInvalidNestedRecursiveAggregate 1622
6.337 rsRecursiveAggregateOfAggregate 1622
6.338 rsPostSortAggregateInAggregateExpression 1622
6.339 rsRunningValueInAggregateExpression 1623
6.340 rsPreviousInAggregateExpression 1623
6.341 rsNestedAggregateViaLookup 1623
6.342 rsNestedAggregateInPageSection 1623
6.343 rsNestedAggregateInFilterExpression 1623
6.344 rsNestedAggregateInGroupVariable 1623
6.345 rsVariableTypeNotSerializable 1623
6.346 rsInvalidWritableVariable 1624
6.347 rsOverallPageNumberInScopedAggregate 1624
6.348 rsOverallPageNumberInBody 1624
6.349 rsParameterPropertyTypeMismatch 1624
6.350 rsStateIndicatorInvalidTransformationScope 1624
6.351 rsVariableInJoinExpression 1624
6.352 rsReportItemInJoinExpression 1624
6.353 rsRunningValueInJoinExpression 1624
6.354 rsPreviousAggregateInJoinExpression 1625
6.355 rsAggregateInJoinExpression 1625
6.356 rsElementMustContainChildren 1625
6.357 rsElementMustContainChild 1625
6.358 rsMissingDefaultRelationshipJoinCondition 1625
6.359 rsNonExistingRelationshipRelatedScope 1625
6.360 rsInvalidSelfJoinRelationship 1625
6.361 rsInvalidDefaultRelationshipNotNaturalJoin 1625
6.362 rsInvalidRelationshipGroupingContainerNotNaturalGroup 1626
6.363 rsInvalidRelationshipContainerNotNaturalJoin 1626
6.364 rsInvalidDefaultRelationshipDuplicateRelatedDataset 1626
6.365 rsInvalidDefaultRelationshipCircularReference 1626
6.366 rsInvalidRelationshipDataSetUsedMoreThanOnce 1626
6.367 rsInvalidRelationshipDataSet 1626
6.368 rsInvalidNaturalSortContainer 1626
6.369 rsInvalidSortingContainerNotNaturalSort 1627
6.370 rsConflictingNaturalSortRequirements 1627
6.371 rsIncompatibleNaturalSortAndNaturalGroup 1627
6.372 rsInvalidNaturalSortFlagCombination 1627
6.373 rsInvalidGroupingNaturalGroupFeature 1627
6.374 rsInvalidGroupingContainerNotNaturalGroup 1627
6.375 rsConflictingNaturalGroupRequirements 1627
6.376 rsInvalidBandInvalidLayoutDirection 1628
6.377 rsInvalidBandPageBreakIsSet 1628
6.378 rsInvalidBandShouldNotBeTogglable 1628
6.379 rsInvalidBandNavigationReference 1628
6.380 rsInvalidBandNavigationItem 1628
6.381 rsInvalidBandNavigations 1628
6.382 rsInvalidSliderDataSetReference 1629
6.383 rsInvalidSliderDataSetReferenceField 1629
6.384 rsBandKeepTogetherIgnored 1629
6.385 rsBandIgnoredProperties 1629
6.386 rsCollationAndCollationCultureSpecified 1629
6.387 rsInvalidAggregateIndicatorField 1629
6.388 rsAggregateIndicatorFieldOnCalculatedField 1630
6.389 rsInvalidSortDirectionMustNotBeSpecified 1630

6.390 rsInvalidNaturalCrossJoin 1630
6.391 rsInvalidIntersectionNaturalCrossJoin 1630
6.392 rsMissingIntersectionRelationshipParentScope 1630
6.393 rsInvalidRelationshipDuplicateParentScope 1630
6.394 rsInvalidCellDataSetName 1631
6.395 rsDefaultRelationshipIgnored 1631
6.396 rsMissingIntersectionDataSetName 1631
6.397 rsInvalidRelationshipTopLevelDataRegion 1631
6.398 rsConflictingSortFlags 1631
6.399 rsInvalidSortFlagCombination 1631
6.400 rsInvalidDeferredSortContainer 1631
6.401 rsDuplicateReportSectionName 1632
6.402 rsInvalidFeatureRdlAttribute 1632
6.403 rsSerializableTypeNotSupported 1632
6.404 rsInvalidScopeReference 1632
6.405 rsInvalidScopeCollectionReference 1632
6.406 rsScopeReferenceInComplexExpression 1632
6.407 rsScopeReferenceUsesDataSetMoreThanOnce 1632

7 Appendix C: Product Behavior 1633

8 Change Tracking 1647

9 Index 1648

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. 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, http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-376,%20Second%20Edition,%20Part%202%20-%20Open%20Packaging%20Conventions.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

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

[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

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

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

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

[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, http://www.unicode.org/unicode/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, http://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, http://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, http://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://docs.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://docs.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**.

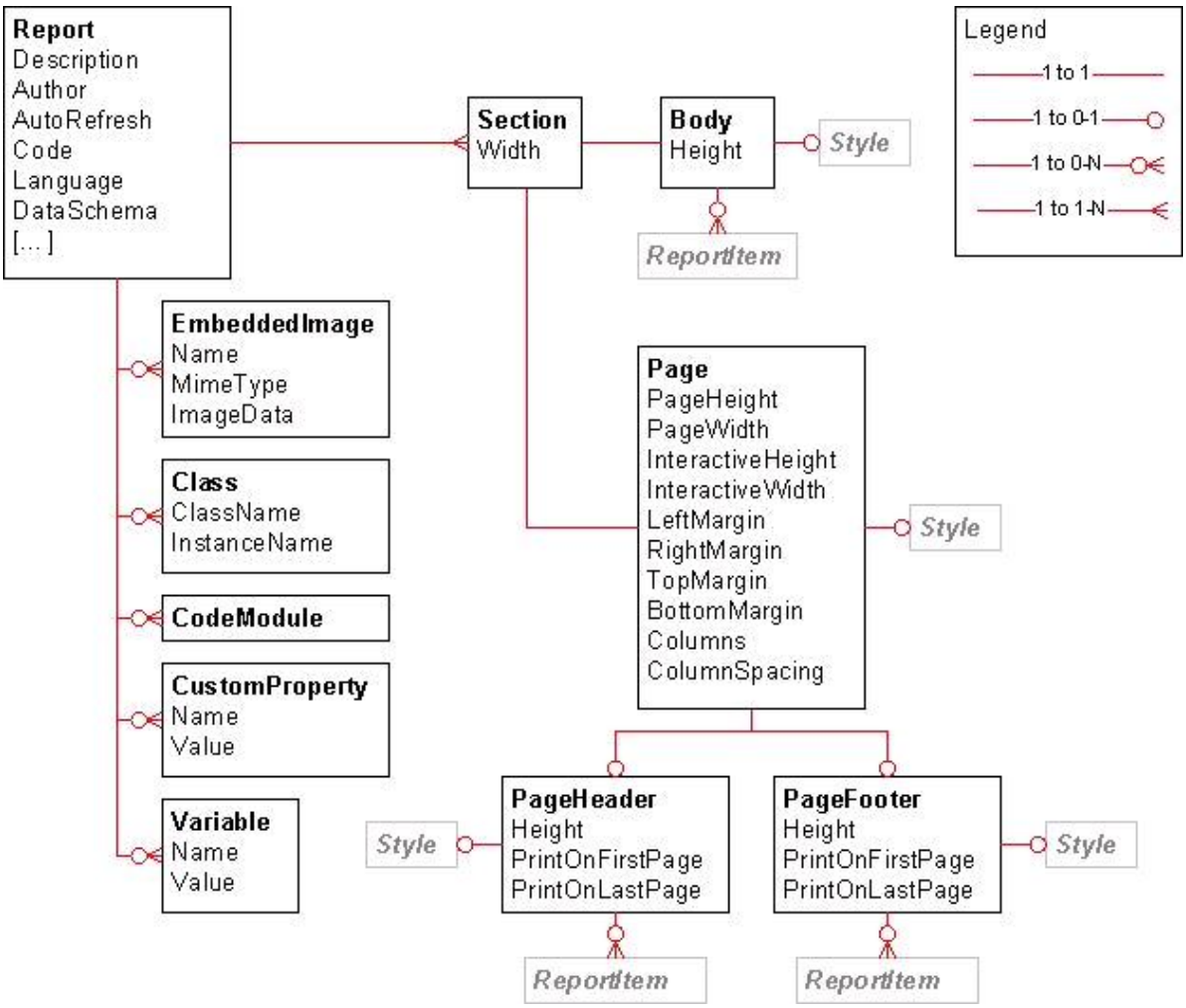

Figure 1: Report layout

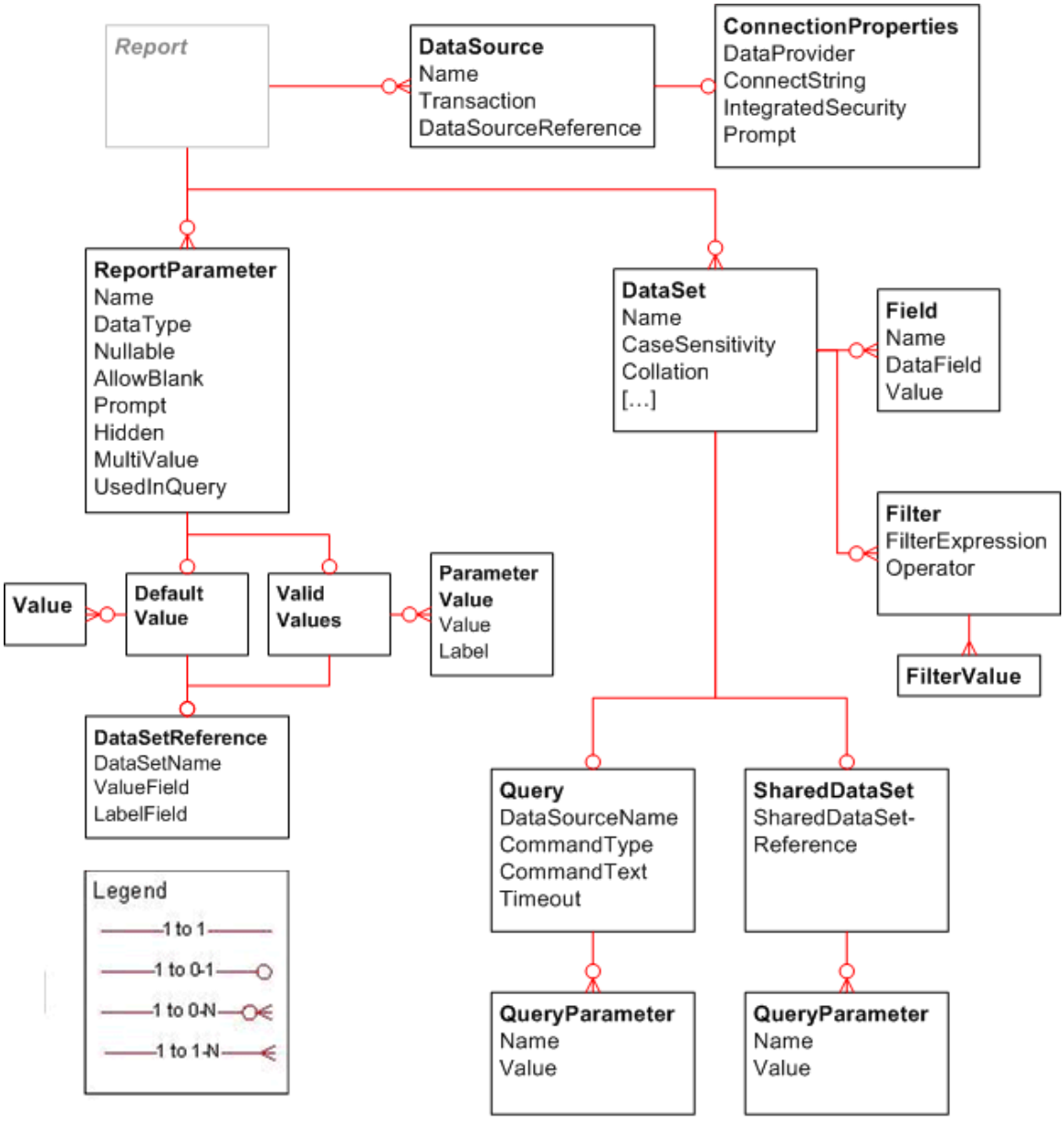

Figure 2: Report data

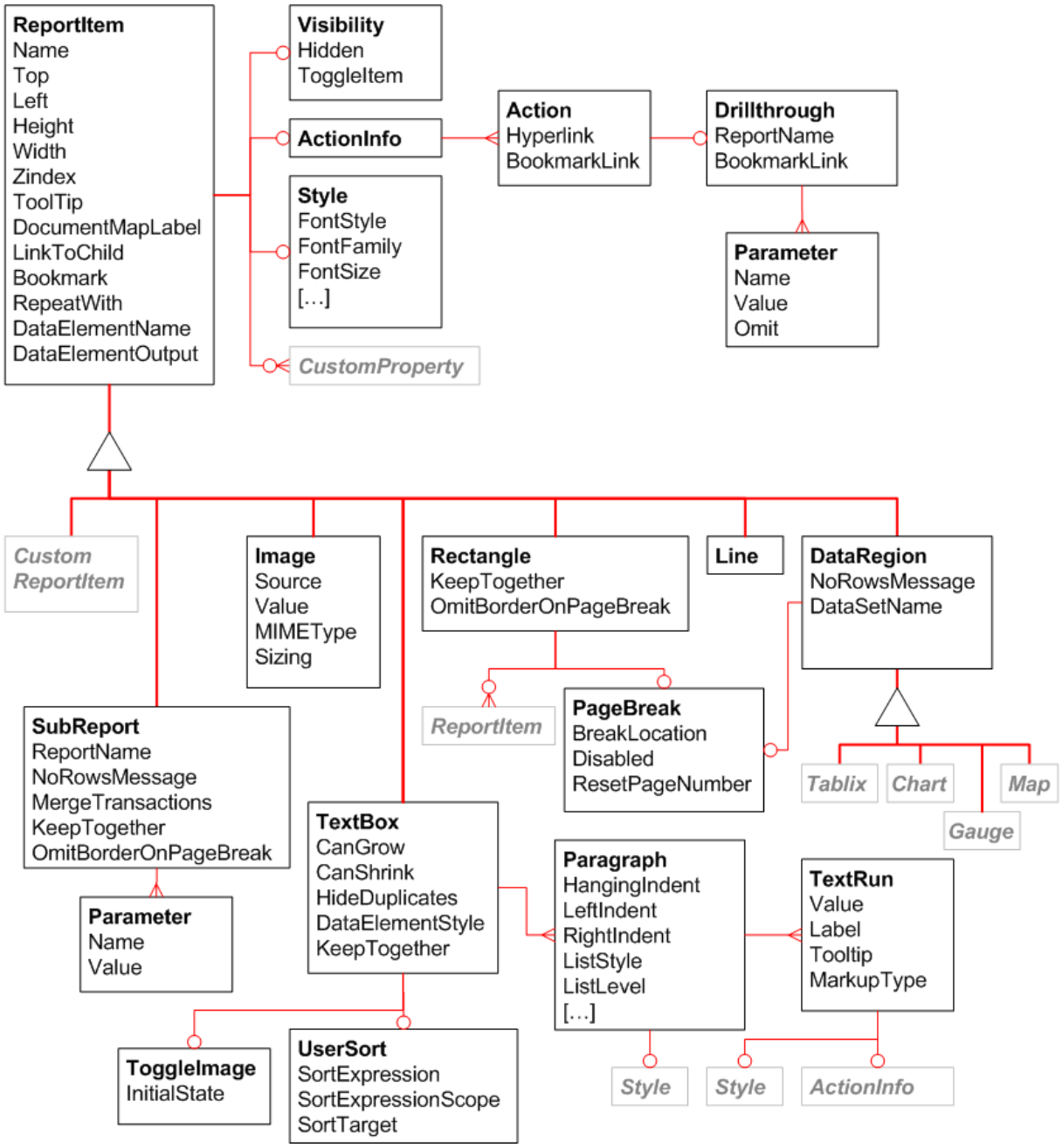

Figure 3: Report items

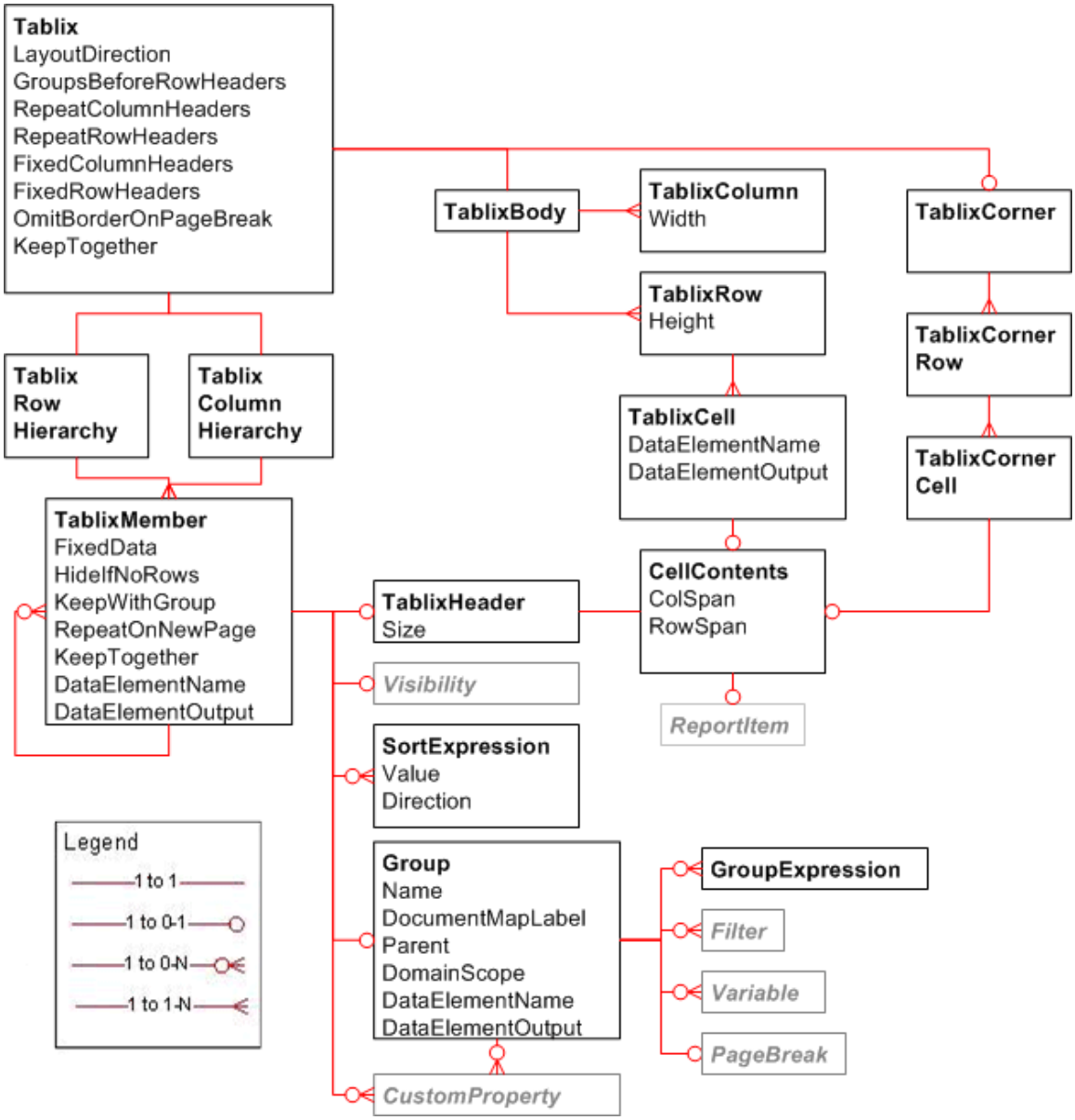

Figure 4: Tablix

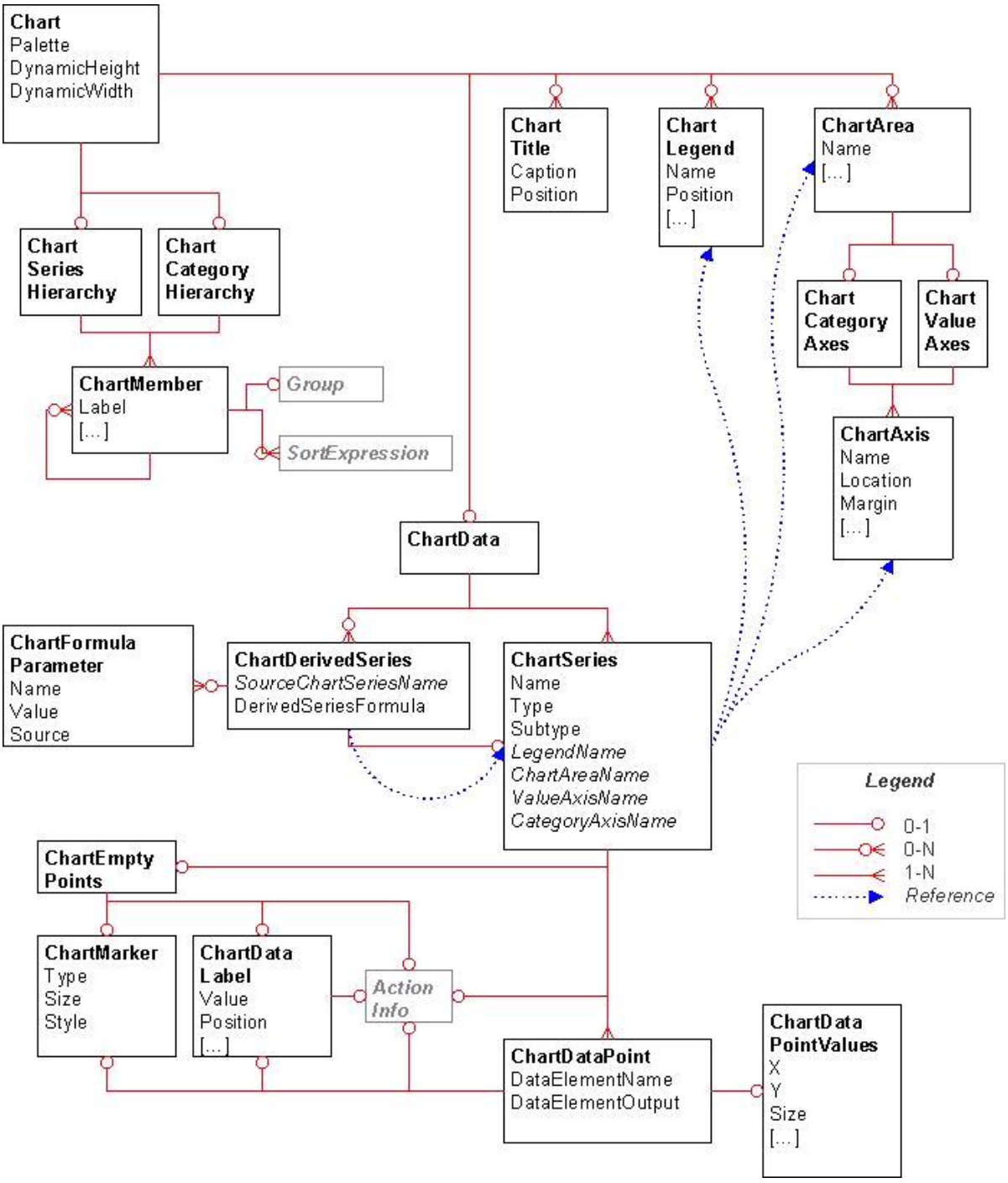

Figure 5: Chart

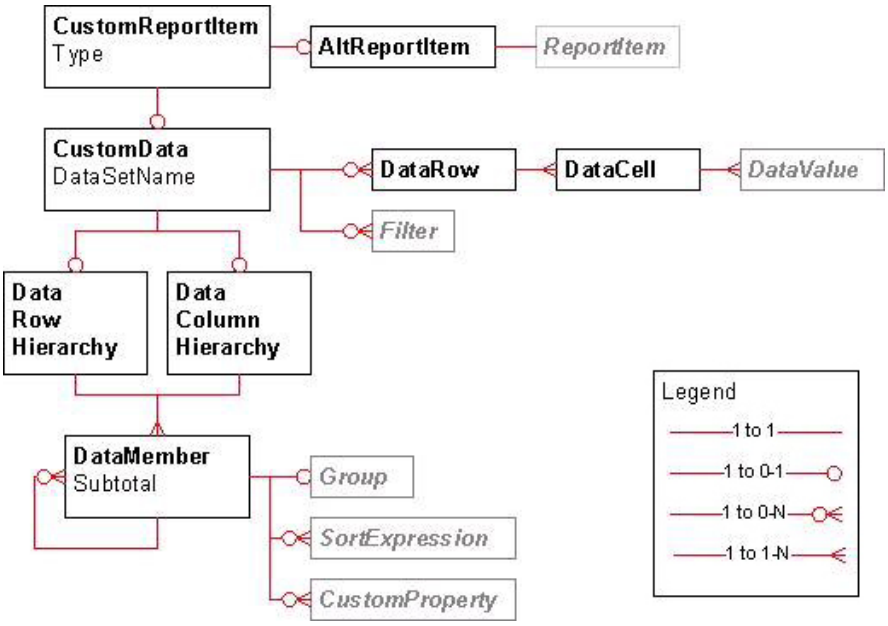

Figure 6: Custom report item

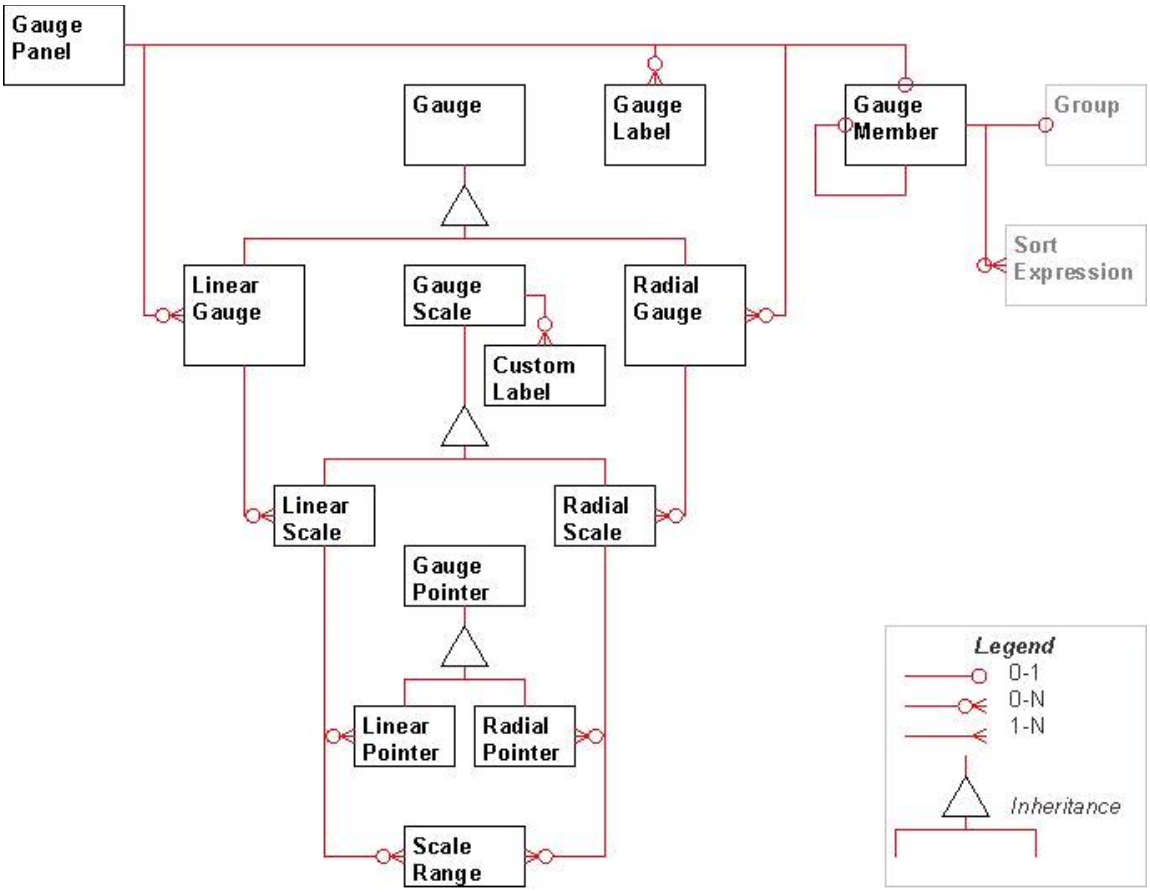

Figure 7: Gauge panel

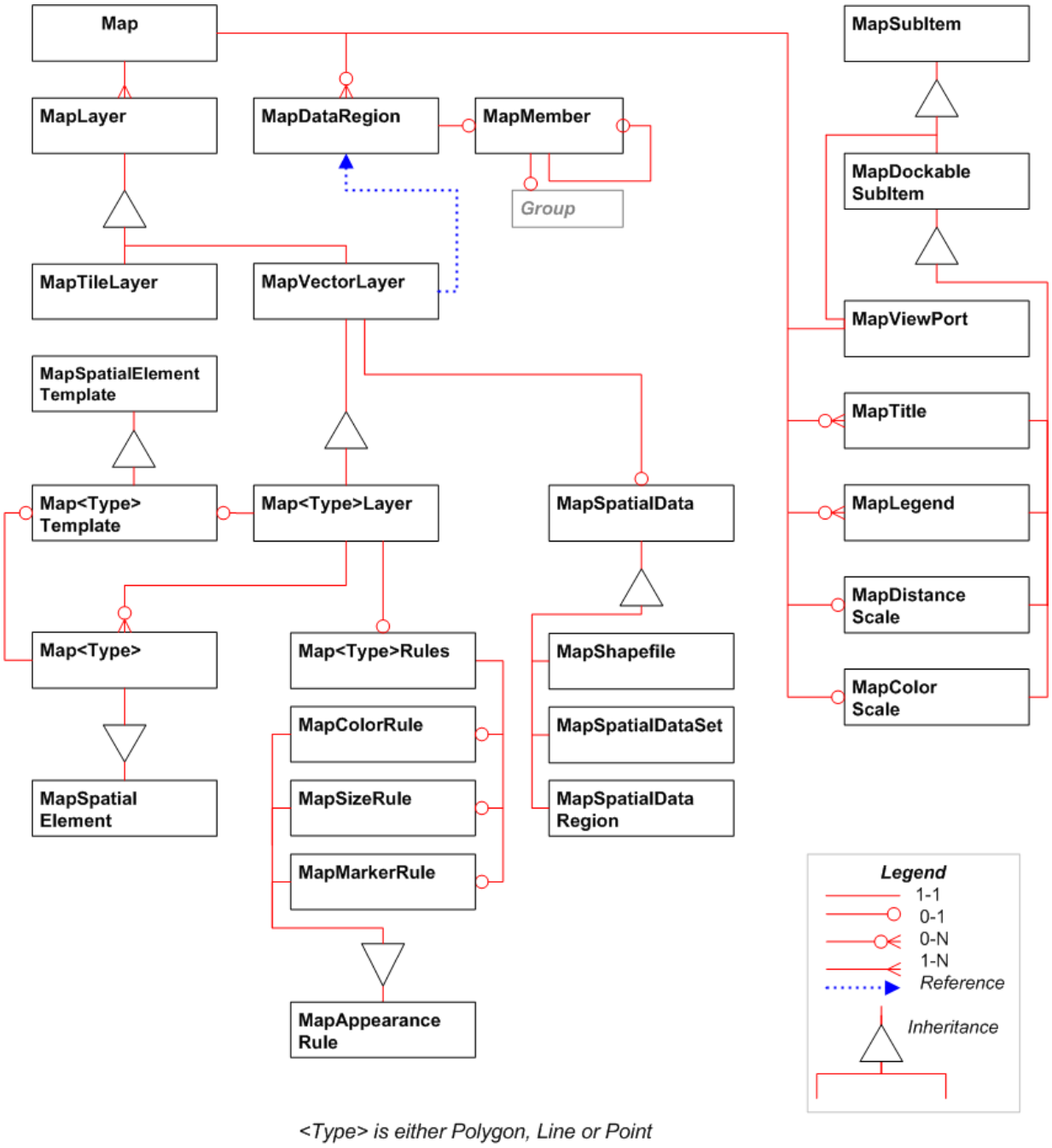

Figure 8: Map structures

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
RdlSize
Image
Line
Textbox
Rectangle
Subreport
Grouping
Grouping.Name
List
List.Name
Matrix
Matrix.Name
Table
Table.Name

Referenced by
Chart
ChartArea
ChartAxis
ChartCodeParameter
ChartLegend
ChartLegendColumn
ChartLegendCustomItem
ChartLegendCustomItemCell
ChartSeries
ChartTitle
CustomReportItem
CustomReportItem.Name
CustomReportItem.Type
DataSet
Field
Group
Group.Name
Tablix
Tablix.Name
CustomLabel
GaugeImage
GaugeLabel
GaugePanel
IndicatorState
LinearGauge
LinearPointer
LinearScale
NumericIndicator
NumericIndicatorRange
RadialGauge
RadialPointer
RadialScale

Referenced by
ScaleRange
StateIndicator
Class
Class.InstanceName
EmbeddedImage
Variable
ReportParameter

2.2.7 RdlSize

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

Elements of the **RdlSize** 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 **RdlSize** is not present, the value of the element is interpreted as

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

where *<FormatType>* is defined above.

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

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

Referenced by
Border.Width

Referenced by
CapImage.OffsetX
CapImage.OffsetY
Image.Height
Image.Left
Image.Top
Image.Width
Line.Height
Line.Left
Line.Top
Line.Width
Page.BottomMargin
Page.ColumnSpacing
Page.InteractiveHeight
Page.InteractiveWidth
Page.LeftMargin
Page.PageHeight
Page.PageWidth
Page.RightMargin
PageSection.Height
PointerImage.OffsetX
PointerImage.OffsetY
Rectangle.Height
Rectangle.Left
Rectangle.Top
Rectangle.Width
Style.LineHeight
Style.PaddingBottom
Style.PaddingLeft
Style.PaddingRight
Style.PaddingTop
Textbox.Height
Textbox.Left

Referenced by
Textbox.Top
Textbox.Width
Paragraph.HangingIndent
Paragraph.RightIndent
Paragraph.SpaceAfter
Paragraph.SpaceBefore
Paragraph.LeftIndent
Style.FontSize
Style.ShadowOffset
GaugeLabel.TextShadowOffset
GaugePanel.Height
GaugePanel.Left
GaugePanel.Top
GaugePanel.Width

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
BackgroundImage.TransparentColor
Border.Color
CapImage.HueColor
CapImage.TransparentColor
FrameImage.HueColor

Referenced by
FrameImage.TransparentColor
GaugeImage.TransparentColor
MapMarkerImage.TransparentColor
PointerImage.HueColor
PointerImage.TransparentColor
ScaleRange.InRangeBarPointerColor
ScaleRange.InRangeTickMarksColor
StateImage.TransparentColor
Style.BackgroundColor
Style.BackgroundGradientEndColor
Style.Color
Style.ShadowColor
TopImage.HueColor
TopImage.TransparentColor

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

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

Color name
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

Color name
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

Color name
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
Action.Hyperlink
BackgroundImage.Value
CapImage.Value
FrameImage.Value
GaugeImage.Value
Image.Value
MapMarkerImage.Value
PointerImage.Value
StateImage.Value
TopImage.Value

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
BackgroundImage.MIMEType
CapImage.MIMEType
EmbeddedImage.MIMEType
FrameImage.MIMEType
GaugeImage.MIMEType
Image.MIMEType
MapMarkerImage.MIMEType
PointerImage.MIMEType
StateImage.MIMEType
TopImage.MIMEType

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
BackgroundImage.Value
CapImage.Value
DataSource.DataSourceReference
Drillthrough.ReportName
FrameImage.Value
GaugeImage.Value
Image.Value
MapMarkerImage.Value
PointerImage.Value

Referenced by
Report.DataTransform
StateImage.Value
Subreport.ReportName
TopImage.Value

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
Report.Language
Style.Language
Style.NumeralLanguage
DataSet.CollationCulture

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 StringWithDataTypeAttribute

The **StringWithDataTypeAttribute** type specifies a String ([XMLSCHEMA2/2] section 3.2.1) value or expression and a **DataType** attribute. The **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**. 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 **DataType** attribute is not present, the value of the **StringWithDataTypeAttribute** type is interpreted as a **String**.

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

Referenced by
Field.Value

Referenced by
FilterValues.FilterValue
QueryParameter.Value
Variable.Value

The following is the XML Schema definition of the **StringWithDataTypeAttribute** type in RDL 2008/01.

```
<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="skip" />
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
```

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

```
<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>
```

2.2.15 LocIDStringWithDataTypeAttribute

The **LocIDStringWithDataTypeAttribute** type specifies a StringWithDataTypeAttribute with an **EvaluationMode** attribute. The **EvaluationMode** attribute specifies whether the value of the **LocIDStringWithDataTypeAttribute** 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 **LocIDStringWithDataTypeAttribute** type can be referenced by the following elements.

Referenced by
TextRun.Value

The following is the XML Schema definition of the **LocIDStringWithDataTypeAttribute** type in RDL 2008/01.

```
<xsd:complexType name="LocIDStringWithDataTypeAttribute">
    <xsd:simpleContent>
      <xsd:extension base="StringWithDataTypeAttribute">
        <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 **LocIDStringWithDataTypeAttribute** type in RDL 2010/01 and RDL 2016/01.

```
<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>
```

2.2.16 StringWithValueTypeAttribute

Applies to RDL 2012/01

The **StringWithValueTypeAttribute** 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 **StringWithValueTypeAttribute** 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
Style.BackgroundColor
Style.Color
Style.FontFamily

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
Report.MustUnderstand

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

Child elements
Report.Author
Report.AutoRefresh
Report.Body
Report.Classes

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

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.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
Report

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

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

2.3.3 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
Report

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" />
```

2.3.4 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
Report

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

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

2.3.5 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
Report

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

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

2.3.6 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" />
```

2.3.7 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" />
```

2.3.8 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
Report

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

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

2.3.9 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.

2.3.10 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
Report

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

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

2.3.11 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
Report

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>
```

2.3.12 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
Report

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

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

2.3.13 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" />
```

2.3.14 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" />
```

2.3.15 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
Report

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

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

2.3.16 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
Report

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" />
```

2.3.17 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" />
```

2.3.18 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
Report

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

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

2.3.19 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
Report

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

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

2.3.20 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
Report

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

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

2.3.21 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" />
```

2.3.22 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" />
```

2.3.23 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" />
```

2.3.24 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" />
```

2.3.25 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" />
```

2.3.26 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" />
```

2.3.27 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" />
```

2.3.28 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" />
```

2.3.29 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" />
```

2.3.30 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" />
```

2.3.31 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" />
```

2.3.32 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" />
```

2.3.33 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" />
```

2.3.34 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" />
```

2.3.35 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" />
```

2.3.36 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" />
```

2.3.37 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 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.

Parent elements
Report

Child elements
ReportSections.ReportSection

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>
```

2.4.1 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
ReportSections

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

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

2.5 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
ReportSections

Child elements
ReportSection.Body
ReportSection.Page
ReportSection.Width
ReportSection.DataElementName
ReportSection.DataElementOutput

Applies to RDL 2011/01

Attributes
ReportSection.Name

Applies to RDL 2012/01

Child elements
ReportSection.LayoutDirection

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>
```

2.5.1 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" />
```

2.5.2 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" />
```

2.5.3 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" />
```

2.5.4 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" />
```

2.5.5 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>
```

2.5.6 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" />
```

2.5.7 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".

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>
```

2.6 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
Report
ReportSection

Child elements
Body.Columns
Body.ColumnSpacing
Body.Style
Body.Height
Body.ReportItems

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>
```

2.6.1 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
Body

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

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

2.6.2 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
Body

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

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

2.6.3 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" />
```

2.6.4 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" />
```

2.6.5 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" />
```

2.7 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
Report
ReportSection

Child elements
Page.Columns
Page.ColumnSpacing

Child elements
Page.BottomMargin
Page.InteractiveHeight
Page.InteractiveWidth
Page.LeftMargin
Page.PageFooter
Page.PageHeader
Page.PageHeight
Page.PageWidth
Page.RightMargin
Page.Style
Page.TopMargin

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: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>
```

2.7.1 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
Page

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

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

2.7.2 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
Page

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

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

2.7.3 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
Page

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

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

2.7.4 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
Page

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

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

2.7.5 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
Page

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

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

2.7.6 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
Page

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

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

2.7.7 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
Page

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

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

2.7.8 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" />
```

2.7.9 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" />
```

2.7.10 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" />
```

2.7.11 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" />
```

2.7.12 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" />
```

2.7.13 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" />
```

2.8 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
Page.PageHeader
Page.PageFooter

Parent elements
Page

Child elements
PageSection.Style
PageSection.Height
PageSection.PrintOnFirstPage
PageSection.PrintOnLastPage
PageSection.PrintBetweenSections
PageSection.ReportItems

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:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

2.8.1 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" />
```

2.8.2 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" />
```

2.8.3 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" />
```

2.8.4 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" />
```

2.8.5 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" />
```

2.8.6 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
PageSection

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

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

2.9 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
Report.PageHeader
Report.PageFooter

Parent elements
Report

Child elements
PageHeaderFooter.Height
PageHeaderFooter.PrintOnFirstPage
PageHeaderFooter.PrintOnLastPage
PageHeaderFooter.ReportItems
PageHeaderFooter.Style

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>
```

2.9.1 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
PageHeaderFooter

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

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

2.9.2 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
PageHeaderFooter

The following is the XML Schema definition of the **PageHeaderFooter.PrintOnFirstPage** element.

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

2.9.3 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
PageHeaderFooter

The following is the XML Schema definition of the **PageHeaderFooter.PrintOnLastPage** element.

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

2.9.4 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
PageHeaderFooter

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

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

2.9.5 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
PageHeaderFooter

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

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

2.10 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
Body
PageSection
PageHeaderFooter
Rectangle
List
MatrixCell
Corner
DynamicColumns
DynamicRows
StaticColumn
StaticRow
Subtotal
TableCell
CustomReportItem

Child elements
ReportItems.Chart
ReportItems.CustomReportItem
ReportItems.GaugePanel
ReportItems.Image
ReportItems.Line
ReportItems.Map
ReportItems.Rectangle
ReportItems.Subreport
ReportItems.Tablix
ReportItems.Textbox
ReportItems.List
ReportItems.Matrix
ReportItems.Table

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: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>
```

2.10.1 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" />
```

2.10.2 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" />
```

2.10.3 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" />
```

2.10.4 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" />
```

2.10.5 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" />
```

2.10.6 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" />
```

2.10.7 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" />
```

2.10.8 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" />
```

2.10.9 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" />
```

2.10.10 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" />
```

2.10.11 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" />
```

2.10.12 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" />
```

2.10.13 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" />
```

2.11 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
ReportItems
CellContents
CustomReportItem.AltReportItem

Attributes
Image.Name

Child elements
Image.Style
Image.ActionInfo
Image.Bookmark

Child elements
Image.CustomProperties
Image.DataElementName
Image.DataElementOutput
Image.DocumentMapLabel
Image.Height
Image.Left
Image.RepeatWith
Image.ToolTip
Image.Top
Image.Visibility
Image.Width
Image.ZIndex
Image.MIMEType
Image.Sizing
Image.Source
Image.Value

Applies to RDL 2011/01

Child elements
Image.Tag

Applies to RDL 2012/01

Child elements
Image.EmbeddingMode

Applies to RDL 2013/01

Child elements
Image.Tags

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:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
  <xsd:anyAttribute namespace="##other" processContents="skip" />
 </xsd:complexType>
```

2.11.1 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" />
```

2.11.2 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" />
```

2.11.3 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.

2.11.4 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
Image

The following is the XML Schema definition of the **Image.Bookmark** element.

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

2.11.5 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
Image

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.

2.11.6 Image.DataElementName

The **Image.DataElementName** element is ignored.

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

Parent elements
Image

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

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

2.11.7 Image.DataElementOutput

The **Image.DataElementOutput** element is ignored.

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

Parent elements
Image

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>
```

2.11.8 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
Image

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" />
```

2.11.9 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" />
```

2.11.10 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" />
```

2.11.11 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" />
```

2.11.12 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" />
```

2.11.13 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" />
```

2.11.14 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" />
```

2.11.15 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" />
```

2.11.16 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" />
```

2.11.17 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" />
```

2.11.18 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>
```

2.11.19 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>
```

2.11.20 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
Image

The following is the XML Schema definition of the **Image.Value** element.

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

2.11.21 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
Image

The following is the XML Schema definition of the **Image.Tag** element.

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

2.11.22 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
Image

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>
```

2.11.23 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
Image

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>
```

2.12 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
ReportItems
CellContents
CustomReportItem.AltReportItem

Attributes
Line.Name

Child elements
Line.Style
Line.ActionInfo
Line.Bookmark
Line.CustomProperties
Line.DataElementName
Line.DataElementOutput
Line.DocumentMapLabel
Line.Height
Line.Left
Line.RepeatWith
Line.ToolTip
Line.Top
Line.Visibility
Line.Width
Line.ZIndex

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>
```

2.12.1 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
Line

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.12.2 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
Line

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

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

2.12.3 Line.ActionInfo

The **Line.ActionInfo** element is ignored.

Following is the parent element of the **Line.ActionInfo** element.

Parent elements
Line

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.

2.12.4 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
Line

The following is the XML Schema definition of the **Line.Bookmark** element.

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

2.12.5 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
Line

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.

2.12.6 Line.DataElementName

The **Line.DataElementName** element is ignored.

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

Parent elements
Line

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

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

2.12.7 Line.DataElementOutput

The **Line.DataElementOutput** element is ignored.

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

Parent elements
Line

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>
```

2.12.8 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
Line

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" />
```

2.12.9 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 0pt. 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
Line

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

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

2.12.10 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
Line

The following is the XML Schema definition of the **Line.Left** element.

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

2.12.11 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
Line

The following is the XML Schema definition of the **Line.RepeatWith** element.

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

2.12.12 Line.ToolTip

The **Line.ToolTip** element is ignored.

Following is the parent element of the **Line.ToolTip** element.

Parent elements
Line

The following is the XML Schema definition of the **Line.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

2.12.13 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
Line

The following is the XML Schema definition of the **Line.Top** element.

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

2.12.14 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
Line

The following is the XML Schema definition of the **Line.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

2.12.15 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 RdlSize. This element can have a negative **RdlSize**.

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
Line

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

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

2.12.16 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 ([XMLSCHEMA2/2] 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 **Line.ZIndex** element.

Parent elements
Line

The following is the XML Schema definition of the **Line.ZIndex** element.

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

2.13 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
ReportItems
CellContents
CustomReportItem.AltReportItem

Attributes
Rectangle.Name

Child elements
Rectangle.Style
Rectangle.ActionInfo
Rectangle.Bookmark
Rectangle.CustomProperties
Rectangle.DataElementName
Rectangle.DataElementOutput

Child elements
Rectangle.DocumentMapLabel
Rectangle.Height
Rectangle.Left
Rectangle.RepeatWith
Rectangle.ToolTip
Rectangle.Top
Rectangle.Visibility
Rectangle.Width
Rectangle.ZIndex
Rectangle.KeepTogether
Rectangle.LinkToChild
Rectangle.OmitBorderOnPageBreak
Rectangle.PageName
Rectangle.PageBreak
Rectangle.ReportItems
Rectangle.PageBreakAtEnd
Rectangle.PageBreakAtStart

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: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>
```

2.13.1 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" />
```

2.13.2 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
Rectangle

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

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

2.13.3 Rectangle.ActionInfo

The **Rectangle.ActionInfo** element is ignored.

Following is the parent element of the **Rectangle.ActionInfo** element.

Parent elements
Rectangle

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.

2.13.4 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
Rectangle

The following is the XML Schema definition of the **Rectangle.Bookmark** element.

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

2.13.5 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.

2.13.6 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" />
```

2.13.7 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>
```

2.13.8 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" />
```

2.13.9 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" />
```

2.13.10 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" />
```

2.13.11 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" />
```

2.13.12 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" />
```

2.13.13 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
Rectangle

The following is the XML Schema definition of the **Rectangle.Top** element.

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

2.13.14 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
Rectangle

The following is the XML Schema definition of the **Rectangle.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

2.13.15 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
Rectangle

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

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

2.13.16 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" />
```

2.13.17 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" " />
```

2.13.18 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" />
```

2.13.19 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" />
```

2.13.20 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" />
```

2.13.21 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" />
```

2.13.22 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
Rectangle

The following is the XML Schema definition of the **Rectangle.PageBreakAtEnd** element.

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

2.13.23 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
Rectangle

The following is the XML Schema definition of the **Rectangle.PageBreakAtStart** element.

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

2.13.24 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
Rectangle

The following is the XML Schema definition of the **Rectangle.PageName** element.

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

2.14 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
ReportItems
CellContents
CustomReportItem.AltReportItem

Attributes
Subreport.Name

Child elements
Subreport.Style
Subreport.ActionInfo
Subreport.Bookmark
Subreport.CustomProperties
Subreport.DataElementName
Subreport.DataElementOutput
Subreport.DocumentMapLabel
Subreport.Height
Subreport.Left
Subreport.LinkToChild
Subreport.RepeatWith
Subreport.ToolTip
Subreport.Top
Subreport.Visibility
Subreport.Width
Subreport.ZIndex
Subreport.KeepTogether
Subreport.MergeTransactions
Subreport.NoRowsMessage

Child elements
Subreport.OmitBorderOnPageBreak
Subreport.Parameters
Subreport.ReportName
Subreport.NoRows

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: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 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: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>
```

2.14.1 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" />
```

2.14.2 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"/>
```

2.14.3 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.

2.14.4 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
Subreport

The following is the XML Schema definition of the **Subreport.Bookmark** element.

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

2.14.5 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
Subreport

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.

2.14.6 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" />
```

2.14.7 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>
```

2.14.8 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" />
```

2.14.9 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" />
```

2.14.10 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
Subreport

The following is the XML Schema definition of the **Subreport.Left** element.

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

2.14.11 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
Subreport

The following is the XML Schema definition of the **Subreport.LinkToChild** element.

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

2.14.12 Subreport.RepeatWith

The **Subreport.RepeatWith** element MUST NOT be specified.

Following is the parent element of the **Subreport.RepeatWith** element.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.RepeatWith** element.

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

2.14.13 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" />
```

2.14.14 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.

Parent elements
Tablix

The following is the XML Schema definition of the **Subreport.Top** element.

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

2.14.15 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.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0"/>
```

2.14.16 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.

Parent elements
Subreport

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

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

2.14.17 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.

Parent elements
Subreport

The following is the XML Schema definition of the **Subreport.ZIndex** element.

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

2.14.18 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" />
```

2.14.19 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" />
```

2.14.20 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" />
```

2.14.21 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" />
```

2.14.22 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" />
```

2.14.23 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" />
```

2.14.24 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" />
```

2.15 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
ReportItems
CellContents
CustomReportItem.AltReportItem

Attributes
Textbox.Name

Child elements
Textbox.Style
Textbox.ActionInfo
Textbox.Bookmark
Textbox.CustomProperties
Textbox.DataElementName
Textbox.DataElementOutput
Textbox.DocumentMapLabel
Textbox.Height
Textbox.Left
Textbox.RepeatWith
Textbox.ToolTip
Textbox.Top
Textbox.Visibility
Textbox.Width

Child elements
Textbox.ZIndex
Textbox.CanGrow
Textbox.CanShrink
Textbox.DataElementStyle
Textbox.HideDuplicates
Textbox.KeepTogether
Textbox.Paragraphs
Textbox.ToggleImage
Textbox.UserSort
Textbox.Value

Applies to RDL 2011/01

Child elements
Textbox.CanScrollVertically

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: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: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: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 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" />
```

2.15.2 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" />
```

2.15.3 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.

2.15.4 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" />
```

2.15.5 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" />
```

2.15.6 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.

2.15.7 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" />
```

2.15.8 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>
```

2.15.9 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" />
```

2.15.10 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" />
```

2.15.11 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" />
```

2.15.12 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" />
```

2.15.13 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" />
```

2.15.14 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" />
```

2.15.15 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" />
```

2.15.16 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" />
```

2.15.17 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 ([XMLSCHEMA2/2] 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" />
```

2.15.18 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" />
```

2.15.19 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" />
```

2.15.20 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 **TextBox.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.

Parent elements
Textbox

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>
```

2.15.21 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.

Parent elements
Textbox

The following is the XML Schema definition of the **Textbox.HideDuplicates** element.

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

2.15.22 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" />
```

2.15.23 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" />
```

2.15.24 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" />
```

2.15.25 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" />
```

2.15.26 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" />
```

2.16 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
Textbox

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>
```

2.16.1 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
Paragraphs

The following is the XML Schema definition of the **Paragraphs.Paragraph** element.

```
<xsd:element name="Paragraph" type="ParagraphType" minOccurs="1" maxOccurs="unbounded"/>
```

2.17 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
Paragraphs

Child elements
Paragraph.HangingIndent
Paragraph.LeftIndent
Paragraph.ListLevel
Paragraph.ListStyle
Paragraph.RightIndent
Paragraph.SpaceAfter
Paragraph.SpaceBefore
Paragraph.Style
Paragraph.TextRuns

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>
```

2.17.1 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" />
```

2.17.2 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" />
```

2.17.3 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" />
```

2.17.4 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>
```

2.17.5 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" />
```

2.17.6 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" />
```

2.17.7 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" />
```

2.17.8 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" />
```

2.17.9 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" />
```

2.18 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
Paragraph

Child elements
TextRuns.TextRun

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>
```

2.18.1 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
TextRuns

The following is the XML Schema definition of the **TextRuns.TextRun** element.

```
<xsd:element name="TextRun" type="TextRunType" minOccurs="1" maxOccurs="unbounded" />
```

2.19 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
TextRuns

Child elements
TextRun.ActionInfo
TextRun.Label
TextRun.MarkupType
TextRun.Style
TextRun.ToolTip
TextRun.Value

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="LocIDStringWithDataTypeAttribute" 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="LocIDStringWithDataTypeAttribute" 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>
```

2.19.1 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" />
```

2.19.2 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" />
```

2.19.3 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
TextRun

The following is the XML Schema definition of the **TextRun.MarkupType** element.

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

2.19.4 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" />
```

2.19.5 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" />
```

2.19.6 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
TextRun.Value.DataType
TextRun.Value.EvaluationMode

The following is the XML Schema definition of the **TextRun.Value** element.

```
<xsd:element name="Value" type="LocIDStringWithDataTypeAttribute" minOccurs="1" />
```

2.19.7 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>
```

2.19.8 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" />
```

2.20 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
ToggleImage.InitialState

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>
```

2.20.1 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">
```

2.21 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
UserSort.SortExpression
UserSort.SortExpressionScope
UserSort.SortTarget

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>
```

2.21.1 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 ([XMLSCHEMA2/2] 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
UserSort

The following is the XML Schema definition of the **UserSort.SortExpression** element.

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

2.21.2 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
UserSort

The following is the XML Schema definition of the **UserSort.SortExpressionScope** element.

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

2.21.3 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
UserSort

The following is the XML Schema definition of the **UserSort.SortTarget** element.

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

2.22 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
ReportItems
CellContents
CustomReportItem.AltReportItem

Attributes
Tablix.Name

Child elements
Tablix.ActionInfo
Tablix.Bookmark
Tablix.CustomProperties
Tablix.DataElementName
Tablix.DataElementOutput
Tablix.DataSetName
Tablix.DocumentMapLabel
Tablix.Filters
Tablix.FixedColumnHeaders

Child elements
Tablix.FixedRowHeaders
Tablix.GroupsBeforeRowHeaders
Tablix.Height
Tablix.KeepTogether
Tablix.LayoutDirection
Tablix.Left
Tablix.NoRowsMessage
Tablix.OmitBorderOnPageBreak
Tablix.PageBreak
Tablix.PageName
Tablix.RepeatColumnHeaders
Tablix.RepeatRowHeaders
Tablix.RepeatWith
Tablix.SortExpressions
Tablix.Style
Tablix.TablixBody
Tablix.TablixColumnHierarchy
Tablix.TablixCorner
Tablix.TablixRowHierarchy
Tablix.ToolTip
Tablix.Top
Tablix.Visibility
Tablix.Width
Tablix.ZIndex

Applies to RDL 2011/01

Child elements
Tablix.BandLayoutOptions
Tablix.BottomMargin
Tablix.CanScroll
Tablix.LeftMargin
Tablix.Relationship

Child elements
Tablix.ReportSlicerState
Tablix.RightMargin
Tablix.TopMargin

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: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: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>
```

2.22.1 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
Tablix

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.22.2 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
Tablix

The following is the XML Schema definition of the **Tablix.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.22.3 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

2.22.4 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

2.22.5 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" />
```

2.22.6 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
Tablix

The following is the XML Schema definition of the **Tablix.CanScroll** element.

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

2.22.7 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
Tablix

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

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

2.22.8 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
Tablix

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

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

2.22.9 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>
```

2.22.10 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" />
```

2.22.11 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" />
```

2.22.12 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" />
```

2.22.13 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" />
```

2.22.14 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" />
```

2.22.15 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" />
```

2.22.16 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" />
```

2.22.17 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" />
```

2.22.18 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>
```

2.22.19 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" />
```

2.22.20 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

2.22.21 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" />
```

2.22.22 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" />
```

2.22.23 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" />
```

2.22.24 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
Tablix

The following is the XML Schema definition of the **Tablix.PageName** element.

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

2.22.25 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
Tablix

The following is the XML Schema definition of the **Tablix.Relationship** element

```
<xsd:element name ="Relationship" type="RelationshipType" minOccurs="0" />
```

2.22.26 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
Tablix

The following is the XML Schema definition of the **Tablix.RepeatColumnHeaders** element.

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

2.22.27 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
Tablix

The following is the XML Schema definition of the **Tablix.RepeatRowHeaders** element.

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

2.22.28 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
Tablix

The following is the XML Schema definition of the **Tablix.RepeatWith** element.

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

2.22.29 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.

2.22.30 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

2.22.31 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" />
```

2.22.32 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" />
```

2.22.33 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" />
```

2.22.34 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" />
```

2.22.35 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" />
```

2.22.36 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" />
```

2.22.37 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" />
```

2.22.38 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" />
```

2.22.39 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

2.22.40 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" />
```

2.22.41 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" />
```

2.22.42 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
Tablix

The following is the XML Schema definition of the **Tablix.ZIndex** element.

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

2.23 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
Tablix

Child elements
TablixBody.TablixColumns
TablixBody.TablixRows

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>
```

2.23.1 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
TablixBody

The following is the XML Schema definition of the **TablixBody.TablixColumns** element.

```
<xsd:element name="TablixColumns" type="TablixColumnsType" minOccurs="1" maxOccurs="1" />
```

2.23.2 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
TablixBody

The following is the XML Schema definition of the **TablixBody.TablixRows** element.

```
<xsd:element name="TablixRows" type="TablixRowsType" minOccurs="1" maxOccurs="1" />
```

2.24 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.

Parent elements
TablixBody

Child elements
TablixColumns.TablixColumn

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>
```

2.24.1 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.

Parent elements
TablixColumns

The following is the XML Schema definition of the **TablixColumns.TablixColumn** element.

```
<xsd:element name="TablixColumn" type="TablixColumnType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.25 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
TablixColumns

Child elements
TablixColumn.Width

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 2010/01and 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>
```

2.25.1 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
TablixColumn

The following is the XML Schema definition of the TablixColumns element.

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

2.26 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
TablixBody

Child elements
TablixRows.TablixRow

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>
```

2.26.1 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
TablixRows

The following is the XML Schema definition of the **TablixRows.TablixRow** element.

```
<xsd:element name="TablixRow" type="TablixRowType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.27 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
TablixRows

Child elements
TablixRow.Height
TablixRow.TablixCells

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>
```

2.27.1 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.

Parent elements
TablixRow

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

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

2.27.2 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.

Parent elements
TablixRow

The following is the XML Schema definition of the **TablixRow.TablixCells** element.

```
<xsd:element name="TablixCells" type="TablixCellsType" minOccurs="1"
             maxOccurs="1" />
```

2.28 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.

Parent elements
TablixRow

Child elements
TablixCells.TablixCell

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>
```

2.28.1 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
TablixCells

The following is the XML Schema definition of the **TablixCells.TablixCell** element.

```
<xsd:element name="TablixCell" type="TablixCellType" minOccurs="1" maxOccurs="unbounded" />
```

2.29 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
TablixCells

Child elements
TablixCell.CellContents
TablixCell.DataElementName
TablixCell.DataElementOutput

Applies to RDL 2011/01

Child elements
TablixCell.DataSetName
TablixCell.Relationships

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 2010/01and 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>
```

2.29.1 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" />
```

2.29.2 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" />
```

2.29.3 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>
```

2.29.4 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" />
```

2.29.5 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
TablixCell

The following is the XML Schema definition of the **TablixCell.Relationships** element.

```
<xsd:element name ="Relationships" type="RelationshipsType" minOccurs="0" />
```

2.30 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
TablixCell
TablixCornerCell
TablixHeader

Child elements
CellContents.Chart
CellContents.ColSpan
CellContents.CustomReportItem
CellContents.GaugePanel
CellContents.Image
CellContents.Line
CellContents.Map
CellContents.Rectangle
CellContents.RowSpan
CellContents.Subreport
CellContents.Tablix
CellContents.Textbox

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: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>
```

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: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>
```

2.30.1 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
CellContents

The following is the XML Schema definition of the **CellContents.Chart** element.

```
<xsd:element name="Chart" type="ChartType" minOccurs="0"
             maxOccurs="1" />
```

2.30.2 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
CellContents

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>
```

2.30.3 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
CellContents

The following is the XML Schema definition of the **CellContents.CustomReportItem** element.

```
<xsd:element name="CustomReportItem" type="CustomReportItemType" minOccurs="0"
             maxOccurs="1" />
```

2.30.4 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
CellContents

The following is the XML Schema definition of the **CellContents.GaugePanel** element.

```
<xsd:element name="GaugePanel" type="GaugePanelType" minOccurs="0" maxOccurs="1" />
```

2.30.5 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
CellContents

The following is the XML Schema definition of the **CellContents.Image** element.

```
<xsd:element name="Image" type="ImageType" minOccurs="0" maxOccurs="1" />
```

2.30.6 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
CellContents

The following is the XML Schema definition of the **CellContents.Line** element.

```
<xsd:element name="Line" type="LineType" minOccurs="0" maxOccurs="1" />
```

2.30.7 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
CellContents

The following is the XML Schema definition of the **CellContents.Map** element.

```
<xsd:element name="Map" type="MapType" minOccurs="0" maxOccurs="1" />
```

2.30.8 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
CellContents

The following is the XML Schema definition of the **CellContents.Rectangle** element.

```
<xsd:element name="Rectangle" type="RectangleType" minOccurs="0" maxOccurs="1" />
```

2.30.9 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
CellContents

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>
```

2.30.10 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
CellContents

The following is the XML Schema definition of the **CellContents.Subreport** element.

```
<xsd:element name="Subreport" type="SubreportType" minOccurs="0" maxOccurs="1" />
```

2.30.11 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
CellContents

The following is the XML Schema definition of the **CellContents.Tablix** element.

```
<xsd:element name="Tablix" type="TablixType" minOccurs="0" maxOccurs="1" />
```

2.30.12 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
CellContents

The following is the XML Schema definition of the **CellContents.Textbox** element.

```
<xsd:element name="Textbox" type=" TextboxType" minOccurs="0" maxOccurs="1" />
```

2.31 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
Tablix

Child elements
TablixColumnHierarchy.TablixMembers

Applies to RDL 2012/01

The following is an additional child element of the **TablixColumnHierarchy** element.

Child elements
TablixColumnHierarchy.EnableDrilldown

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>
```

2.31.1 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
TablixColumnHierarchy

The following is the XML Schema definition of the **TablixColumnHierarchy.TablixMembers** element.

```
<xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1" maxOccurs="1" />
```

2.31.2 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
TablixColumnHierarchy

The following is the XML Schema definition of the **TablixColumnHierarchy.EnableDrilldown** element.

```
<xsd:element name ="EnableDrilldown" type="xsd:boolean" />
```

2.32 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
TablixColumnHierarchy
TablixMember
TablixRowHierarchy

Child elements
TablixMembers.TablixMember

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>
```

2.32.1 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.

Parent elements
TablixMembers

The following is the XML Schema definition of the **TablixMembers.TablixMember** element.

```
<xsd:element name="TablixMember" type="TablixMemberType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.33 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.

Parent elements
TablixMembers

Child elements
TablixMember.CustomProperties
TablixMember.DataElementName
TablixMember.DataElementOutput
TablixMember.FixedData
TablixMember.Group
TablixMember.HideIfNoRows
TablixMember.KeepTogether
TablixMember.KeepWithGroup
TablixMember.RepeatOnNewPage

Child elements
TablixMember.SortExpressions
TablixMember.TablixHeader
TablixMember.TablixMembers
TablixMember.Visibility

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: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>
```

2.33.1 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" />
```

2.33.2 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" />
```

2.33.3 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>
```

2.33.4 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" />
```

2.33.5 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" />
```

2.33.6 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" />
```

2.33.7 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" />
```

2.33.8 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>
```

2.33.9 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" />
```

2.33.10 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" />
```

2.33.11 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.

Parent elements
TablixMember

The following is the XML Schema definition of the **TablixMember.TablixHeader** element.

```
<xsd:element name="TablixHeader" type="TablixHeaderType" minOccurs="0" maxOccurs="1" />
```

2.33.12 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.

Parent elements
TablixMember

The following is the XML Schema definition of the **TablixMember.TablixMembers** element.

```
<xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="0" maxOccurs="1" />
```

2.33.13 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.

Parent elements
TablixMember

The following is the XML Schema definition of the **TablixMember.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" maxOccurs="1" />
```

2.34 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
TablixMember

Child elements
TablixHeader.CellContents
TablixHeader.Size

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>
```

2.34.1 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" />
```

2.34.2 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
TablixHeader

The following is the XML Schema definition of the **TablixHeader.Size** element.

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

2.35 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
Tablix

Child elements
TablixCorner.TablixCornerRows

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>
```

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>
```

2.35.1 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
TablixCorner

The following is the XML Schema definition of the **TablixCorner.TablixCornerRows** element.

```
<xsd:element name="TablixCornerRows" type="TablixCornerRowsType" minOccurs="1"
             maxOccurs="1" />
```

2.36 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
TablixCorner

Child elements
TablixCornerRows.TablixCornerRow

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>
```

2.36.1 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.

Parent elements
TablixCornerRows

The following is the XML Schema definition of the **TablixCornerRows.TablixCornerRow** element.

```
<xsd:element name="TablixCornerRow" type="TablixCornerRowType" minOccurs="1"
             maxOccurs="unbounded">
</xsd:element>
```

2.37 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.

Parent elements
TablixCornerRows

Child elements
TablixCornerRow.TablixCornerCell

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>
```

2.37.1 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.

Parent elements
TablixCornerRow

The following is the XML Schema definition of the **TablixCornerRow.TablixCornerCell** element.

```
<xsd:element name="TablixCornerCell" type="TablixCornerCellType" minOccurs="0"
             maxOccurs="unbounded">
</xsd:element>
```

2.38 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
TablixCornerRow

Child elements
TablixCornerCell.CellContents

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>
```

2.38.1 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" />
```

2.39 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
TablixHierarchy.TablixMembers

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:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

2.39.1 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
TablixHierarchy

The following is the XML Schema definition of the **TablixHierarchy.TablixMembers** element.

```
<xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1"
             maxOccurs="1" />
```

2.40 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
Tablix

Child elements
TablixRowHierarchy.TablixMembers

Applies to RDL 2012/01

The following is an additional child element of the **TablixRowHierarchy** element.

Child elements
TablixRowHierarchy.EnableDrilldown

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>
```

2.40.1 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.

Parent elements
TablixHierarchy

The following is the XML Schema definition of the **TablixRowHierarchy.TablixMembers** element.

```
<xsd:element name="TablixMembers" type="TablixMembersType" minOccurs="1"
             maxOccurs="1" />
```

2.40.2 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
TablixRowHierarchy

The following is the XML Schema definition of the **TablixRowHierarchy.EnableDrilldown** element.

```
<xsd:element name ="EnableDrilldown" type="xsd:boolean" />
```

2.41 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
Tablix

Child elements
RowCount
ColumnCount
Coverflow
PlayAxis
Tabstrip

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:any namespace="##other" processContents="lax" />
    </xsd:choice>
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
</xsd:element>
```

2.41.1 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.

Parent elements
BandLayoutOptions

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>
```

2.41.2 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.

Parent elements
BandLayoutOptions

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>
```

2.41.3 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
BandLayoutOptions

The following is the XML Schema definition of the **BandLayoutOptions.Coverflow** element.

```
<xsd:element name="Coverflow" type="CoverflowType" minOccurs="0"/>
```

2.41.4 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
BandLayoutOptions

The following is the XML Schema definition of the **BandLayoutOptions.PlayAxis** element.

```
<xsd:element name="PlayAxis" type="PlayAxisType" minOccurs="0"/>
```

2.41.5 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.

Parent elements
BandLayoutOptions

The following is the XML Schema definition of the **BandLayoutOptions.Tabstrip** element.

```
<xsd:element name="Tabstrip" type="TabstripType" minOccurs="0"/>
```

2.42 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.

Parent elements
BandLayoutOptions

Child elements
NavigationItem
Slider

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>
```

2.42.1 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.

Parent elements
Coverflow

The following is the XML Schema definition of the Coverflow.NavigationItem element.

```
<xsd:element name="NavigationItem" type="NavigationItemType" minOccurs="0"  maxOccurs="1"/>
```

2.42.2 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.

Parent elements
Coverflow

The following is the XML Schema definition of the **Coverflow.Slider** element.

```
<xsd:element name="Slider" type="SliderType" minOccurs="0"  maxOccurs="1"/>
```

2.43 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.

Parent elements
BandLayoutOptions

Child elements
NavigationItem
Slider

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>
```

2.43.1 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
Tabstrip

The following is the XML Schema definition of the **Tabstrip.NavigationItem** element.

```
<xsd:element name="NavigationItem" type="NavigationItemType" minOccurs="0" maxOccurs="1"/>
```

2.43.2 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
Tabstrip

The following is the XML Schema definition of the **Tabstrip.Slider** element.

```
<xsd:element name="Slider" type="SliderType" minOccurs="0"  maxOccurs="1"/>
```

2.44 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
BandLayoutOptions

Child elements
Slider
DockingOption

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>
```

2.44.1 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
PlayAxis

The following is the XML Schema definition of the **PlayAxis.Slider** element.

```
<xsd:element name="Slider" type="SliderType" minOccurs="0"  maxOccurs="1"/>
```

2.44.2 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
PlayAxis

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>
```

2.45 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
Coverflow
Tabstrip

Child elements
ReportItemReference
ReportItem

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>
```

2.45.1 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.ReportItemis specified, this element MUST NOT be specified.

The following is the parent element of the **NavigationItem.ReportItemReference** element.

Parent elements
NavigationItem

The following is the XML Schema definition of the **NavigationItem.ReportItemReference** element.

```
<xsd:element name="ReportItemReference" type="xsd:string" minOccurs="0" maxOccurs="1"/>
```

2.45.2 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
NavigationItem

The following is the XML Schema definition of the NavigationItem.ReportItem element.

```
<xsd:element name="ReportItem" type="rdl2010:ReportItemsType" minOccurs="0" maxOccurs="1"/>
```

2.46 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
Coverflow
Tabstrip
PlayAxis

Child elements
Hidden
LabelData

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>
```

2.46.1 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
Slider

The following is the XML Schema definition of the **Slider.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:boolean" minOccurs="0" maxOccurs="1"/>
```

2.46.2 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
Slider

The following is the XML Schema definition of the **Slider.LabelData** element.

```
<xsd:element name="LabelData" type="LabelDataType" minOccurs="0"  maxOccurs="1"/>
```

2.47 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
Slider

Child elements
DataSetName
Key
Label

Applies to RDL 2013/01

Child elements
KeyFields

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>
```

2.47.1 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
LabelData

The following is the XML Schema definition of the **LabelData.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="0" maxOccurs="1"/>
```

2.47.2 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
LabelData

The following is the XML Schema definition of the **LabelData.Key** element.

```
<xsd:element name="Key" type="xsd:string" minOccurs="0" maxOccurs="1"/>
```

2.47.3 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
LabelData

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>
```

2.47.4 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
LabelData

The following is the XML Schema definition of the **LabelData.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0"  maxOccurs="1"/>
```

2.48 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
ReportItems
CustomReportItem.AltReportItem

Attributes
List.Name

Child elements
List.Style
List.Action
List.LinkToChild
List.Bookmark
List.CustomProperties
List.DataElementName
List.DataElementOutput
List.Label
List.Height
List.Left
List.RepeatWith
List.ToolTip
List.Top
List.Visibility
List.Width
List.ZIndex
List.KeepTogether
List.NoRows
List.PageBreakAtEnd
List.PageBreakAtStart
List.DataSetName

Child elements
List.Filters
List.DataInstanceElementOutput
List.DataInstanceName
List.FillPage
List.Grouping
List.ReportItems
List.Sorting

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: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>
```

2.48.1 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" />
```

2.48.2 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
List

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

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

2.48.3 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
List

The following is the XML Schema definition of the **List.Action** element.

```
<xsd:element name="Action" type="ActionType" minOccurs="0" />
```

2.48.4 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
List

The following is the XML Schema definition of the **List.LinkToChild** element.

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

2.48.5 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 actionaction. 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" />
```

2.48.6 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
List

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.

2.48.7 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" />
```

2.48.8 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>
```

2.48.9 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" />
```

2.48.10 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" />
```

2.48.11 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" />
```

2.48.12 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
List

The following is the XML Schema definition of the **List.RepeatWith** element.

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

2.48.13 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" />
```

2.48.14 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" />
```

2.48.15 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" />
```

2.48.16 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" />
```

2.48.17 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" />
```

2.48.18 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" />
```

2.48.19 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" />
```

2.48.20 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" />
```

2.48.21 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" />
```

2.48.22 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" />
```

2.48.23 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" />
```

2.48.24 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>
```

2.48.25 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" />
```

2.48.26 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
List

The following is the XML Schema definition of the **List.FillPage** element.

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

2.48.27 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
List

The following is the XML Schema definition of the **List.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" minOccurs="0" />
```

2.48.28 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
List

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

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

2.48.29 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
List

The following is the XML Schema definition of the **List.Sorting** element.

```
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
```

2.49 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
ReportItems
CustomReportItem.AltReportItem

Attributes
Matrix.Name

Child elements
Matrix.Style

Child elements
Matrix.Action
Matrix.LinkToChild
Matrix.Bookmark
Matrix.CustomProperties
Matrix.DataElementName
Matrix.DataElementOutput
Matrix.Label
Matrix.Height
Matrix.Left
Matrix.RepeatWith
Matrix.ToolTip
Matrix.Top
Matrix.Visibility
Matrix.Width
Matrix.ZIndex
Matrix.KeepTogether
Matrix.NoRows
Matrix.PageBreakAtEnd
Matrix.PageBreakAtStart
Matrix.DataSetName
Matrix.Filters
Matrix.CellDataElementName
Matrix.CellDataElementOutput
Matrix.ColumnGroupings
Matrix.Corner
Matrix.GroupsBeforeRowHeaders
Matrix.LayoutDirection
Matrix.MatrixColumns
Matrix.MatrixRows
Matrix.RowGroupings

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: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>
```

2.49.1 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" />
```

2.49.2 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" />
```

2.49.3 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
Matrix

The following is the XML Schema definition of the **Matrix.Action** element.

```
<xsd:element name="Action" type="ActionType" minOccurs="0" />
```

2.49.4 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
Matrix

The following is the XML Schema definition of the **Matrix.LinkToChild** element.

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

2.49.5 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" />
```

2.49.6 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
Matrix

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.

2.49.7 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" />
```

2.49.8 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>
```

2.49.9 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
Matrix

The following is the XML Schema definition of the **Matrix.Label** element.

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

2.49.10 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
Matrix

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

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

2.49.11 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
Matrix

The following is the XML Schema definition of the **Matrix.Left** element.

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

2.49.12 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
Matrix

The following is the XML Schema definition of the **Matrix.RepeatWith** element.

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

2.49.13 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
Matrix

The following is the XML Schema definition of the **Matrix.ToolTip** element.

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

2.49.14 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
Matrix

The following is the XML Schema definition of the **Matrix.Top** element.

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

2.49.15 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" />
```

2.49.16 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" />
```

2.49.17 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" />
```

2.49.18 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" />
```

2.49.19 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" />
```

2.49.20 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" />
```

2.49.21 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" />
```

2.49.22 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" />
```

2.49.23 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" />
```

2.49.24 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" />
```

2.49.25 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>
```

2.49.26 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
Matrix

The following is the XML Schema definition of the **Matrix.ColumnGroupings** element.

```
<xsd:element name="ColumnGroupings" type="ColumnGroupingsType" />
```

2.49.27 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" />
```

2.49.28 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
Matrix

The following is the XML Schema definition of the **Matrix.GroupsBeforeRowHeaders** element.

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

2.49.29 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
Matrix

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>
```

2.49.30 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
Matrix

The following is the XML Schema definition of the **Matrix.MatrixColumns** element.

```
<xsd:element name="MatrixColumns" type="MatrixColumnsType" />
```

2.49.31 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
Matrix

The following is the XML Schema definition of the **Matrix.MatrixRows** element.

```
<xsd:element name="MatrixRows" type="MatrixRowsType" />
```

2.49.32 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" />
```

2.50 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
ColumnGroupings.ColumnGrouping

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>
```

2.50.1 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" />
```

2.51 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
ColumnGroupings

Child elements
ColumnGrouping.DynamicColumns
ColumnGrouping.FixedHeader
ColumnGrouping.Height
ColumnGrouping.StaticColumns

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>
```

2.51.1 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
ColumnGrouping

The following is the XML Schema definition of the **ColumnGrouping.DynamicColumns** element.

```
<xsd:element name="DynamicColumns" type="DynamicColumnsRowsType" minOccurs="0" />
```

2.51.2 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
ColumnGrouping

The following is the XML Schema definition of the **ColumnGrouping.FixedHeader** element.

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

2.51.3 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
ColumnGrouping

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

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

2.51.4 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
ColumnGrouping

The following is the XML Schema definition of the **ColumnGrouping.StaticColumns** element.

```
<xsd:element name="StaticColumns" type="StaticColumnsType" minOccurs="0" />
```

2.52 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
ColumnGrouping

Child elements
DynamicColumns.Grouping
DynamicColumns.ReportItems
DynamicColumns.Sorting

Child elements
DynamicColumns.Subtotal
DynamicColumns.Visibility

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>
```

2.52.1 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
DynamicColumns

The following is the XML Schema definition of the **DynamicColumns.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" />
```

2.52.2 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
DynamicColumns

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

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

2.52.3 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
DynamicColumns

The following is the XML Schema definition of the **DynamicColumns.Sorting** element.

```
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
```

2.52.4 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
DynamicColumns

The following is the XML Schema definition of the **DynamicColumns.Subtotal** element.

```
<xsd:element name="Subtotal" type="SubtotalType" minOccurs="0" />
```

2.52.5 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
DynamicColumns

The following is the XML Schema definition of the **DynamicColumns.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

2.53 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
DynamicColumns
DynamicRows

Child elements
Subtotal.DataElementName
Subtotal.DataElementOutput
Subtotal.Position
Subtotal.ReportItems
Subtotal.Style

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>
```

2.53.1 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" />
```

2.53.2 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.

Parent elements
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>
```

2.53.3 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.

Parent elements
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>
```

2.53.4 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" />
```

2.53.5 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" />
```

2.54 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

Child elements
StaticColumns.StaticColumn

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>
```

2.54.1 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.

Parent elements
StaticColumns

The following is the XML Schema definition of the **StaticColumns.StaticColumn** element.

```
<xsd:element name="StaticColumn" type="StaticColumnType" maxOccurs="unbounded" />
```

2.55 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.

Parent elements
StaticColumns

Child elements
StaticColumn.ReportItems

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>
```

2.55.1 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
StaticColumn

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

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

2.56 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
Matrix

Child elements
Corner.ReportItems

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>
```

2.56.1 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
Corner

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

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

2.57 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
MatrixColumns.MatrixColumn

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>
```

2.57.1 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
MatrixColumns

The following is the XML Schema definition of the **MatrixColumns.MatrixColumn** element.

```
<xsd:element name="MatrixColumn" type="MatrixColumnType" maxOccurs="unbounded" />
```

2.58 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
MatrixColumns

Child elements
MatrixColumn.Width

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>
```

2.58.1 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
MatrixColumn

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

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

2.59 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
MatrixRows.MatrixRow

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>
```

2.59.1 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" />
```

2.60 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
MatrixRows

Child elements
MatrixRow.Height

Child elements
MatrixRow.MatrixCells

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>
```

2.60.1 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
MatrixRow

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

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

2.60.2 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
MatrixRow

The following is the XML Schema definition of the **MatrixRow.MatrixCells** element.

```
<xsd:element name="MatrixCells" type="MatrixCellsType" />
```

2.61 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
MatrixRow

Child elements
MatrixCells.MatrixCell

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>
```

2.61.1 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" />
```

2.62 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
MatrixCells

Child elements
MatrixCell.ReportItems

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>
```

2.62.1 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
MatrixCell

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

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

2.63 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
RowGroupings.RowGrouping

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>
```

2.63.1 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" />
```

2.64 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
RowGroupings

Child elements
RowGrouping.DynamicRows
RowGrouping.FixedHeader
RowGrouping.StaticRows
RowGrouping.Width

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>
```

2.64.1 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
RowGrouping

The following is the XML Schema definition of the **RowGrouping.DynamicRows** element.

```
<xsd:element name="DynamicRows" type="DynamicColumnsRowsType" minOccurs="0" />
```

2.64.2 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
RowGrouping

The following is the XML Schema definition of the **RowGrouping.FixedHeader** element.

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

2.64.3 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
RowGrouping

The following is the XML Schema definition of the **RowGrouping.StaticRows** element.

```
<xsd:element name="StaticRows" type="StaticRowsType" minOccurs="0" />
```

2.64.4 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
RowGrouping

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

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

2.65 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
RowGrouping

Child elements
DynamicRows.Grouping
DynamicRows.ReportItems
DynamicRows.Sorting
DynamicRows.Subtotal
DynamicRows.Visibility

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>
```

2.65.1 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
DynamicRows

The following is the XML Schema definition of the **DynamicRows.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" />
```

2.65.2 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
DynamicRows

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

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

2.65.3 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
DynamicRows

The following is the XML Schema definition of the **DynamicRows.Sorting** element.

```
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
```

2.65.4 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.

Parent elements
DynamicRows

The following is the XML Schema definition of the **DynamicRows.Subtotal** element.

```
<xsd:element name="Subtotal" type="SubtotalType" minOccurs="0" />
```

2.65.5 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.

Parent elements
DynamicRows

The following is the XML Schema definition of the **DynamicRows.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

2.66 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.

Parent elements
RowGrouping

Child elements
StaticRows.StaticRow

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>
```

2.66.1 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
StaticRows

The following is the XML Schema definition of the **StaticRows.StaticRow** element.

```
<xsd:element name="StaticRow" type="StaticRowType" maxOccurs="unbounded" />
```

2.67 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
StaticRows

Child elements
StaticRow.ReportItems

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>
```

2.67.1 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
StaticRow

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

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

2.68 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
ReportItems
CustomReportItem.AltReportItem

Attributes
Table.Name

Child elements
Table.Style
Table.Action
Table.LinkToChild
Table.Bookmark
Table.CustomProperties
Table.DataElementName
Table.DataElementOutput
Table.Label
Table.Height
Table.Left
Table.RepeatWith
Table.ToolTip
Table.Top
Table.Visibility
Table.Width
Table.ZIndex

Child elements
Table.KeepTogether
Table.NoRows
Table.PageBreakAtEnd
Table.PageBreakAtStart
Table.DataSetName
Table.Filters
Table.DetailDataCollectionName
Table.DetailDataElementName
Table.DetailDataElementOutput
Table.Details
Table.FillPage
Table.Footer
Table.Header
Table.TableColumns
Table.TableGroups

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: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>
```

2.68.1 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" />
```

2.68.2 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
Table

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

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

2.68.3 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
Table

The following is the XML Schema definition of the **Table.Action** element.

```
<xsd:element name="Action" type="ActionType" minOccurs="0" />
```

2.68.4 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
Table

The following is the XML Schema definition of the **Table.LinkToChild** element.

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

2.68.5 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" />
```

2.68.6 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
Table

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.

2.68.7 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" />
```

2.68.8 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>
```

2.68.9 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" />
```

2.68.10 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" />
```

2.68.11 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" />
```

2.68.12 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
Table

The following is the XML Schema definition of the **Table.RepeatWith** element.

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

2.68.13 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" />
```

2.68.14 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" />
```

2.68.15 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" />
```

2.68.16 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" />
```

2.68.17 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" />
```

2.68.18 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
List

The following is the XML Schema definition of the **Table.KeepTogether** element.

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

2.68.19 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" />
```

2.68.20 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" />
```

2.68.21 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" />
```

2.68.22 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" />
```

2.68.23 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.

Parent elements
Table

The following is the XML Schema definition of the **Table.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

2.68.24 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.

Parent elements
Table

The following is the XML Schema definition of the **Table.DetailDataCollectionName** element.

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

2.68.25 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" />
```

2.68.26 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>
```

2.68.27 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" />
```

2.68.28 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
Table

The following is the XML Schema definition of the **Table.FillPage** element.

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

2.68.29 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" />
```

2.68.30 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" />
```

2.68.31 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
Table

The following is the XML Schema definition of the **Table.TableColumns** element.

```
<xsd:element name="TableColumns" type="TableColumnsType" />
```

2.68.32 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
Table

The following is the XML Schema definition of the **Table.TableGroups** element.

```
<xsd:element name="TableGroups" type="TableGroupsType" minOccurs="0" />
```

2.69 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
Table

Child elements
Details.Grouping
Details.Sorting
Details.TableRows

Child elements
Details.Visibility

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>
```

2.69.1 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
Details

The following is the XML Schema definition of the **Details.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" minOccurs="0" />
```

2.69.2 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
Details

The following is the XML Schema definition of the **Details.Sorting** element.

```
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
```

2.69.3 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
Details

The following is the XML Schema definition of the **Details.TableRows** element.

```
<xsd:element name="TableRows" type="TableRowsType" />
```

2.69.4 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
Details

The following is the XML Schema definition of the **Details.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

2.70 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
Details
Header
Footer

Child elements
TableRows.TableRow

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>
```

2.70.1 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
TableRows

The following is the XML Schema definition of the **TableRows.TableRow** element.

```
<xsd:element name="TableRow" type="TableRowType" maxOccurs="unbounded" />
```

2.71 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
TableRows

Child elements
TableRow.Height
TableRow.TableCells
TableRow.Visibility

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>
```

2.71.1 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
TableRow

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

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

2.71.2 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
TableRow

The following is the XML Schema definition of the **TableRow.TableCells** element.

```
<xsd:element name="TableCells" type="TableCellsType" />
```

2.71.3 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
TableRow

The following is the XML Schema definition of the **TableRow.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

2.72 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
TableRow

Child elements
TableCells.TableCell

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>
```

2.72.1 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" />
```

2.73 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
TableCells

Child elements
TableCell.ColSpan
TableCell.ReportItems

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>
```

2.73.1 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 descendents 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
TableCell

The following is the XML Schema definition of the **TableCell.ColSpan** element.

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

2.73.2 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
TableCell

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

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

2.74 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
Table
TableGroup

Child elements
Footer.RepeatOnNewPage
Footer.TableRows

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>
```

2.74.1 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
Footer

The following is the XML Schema definition of the **Footer.RepeatOnNewPage** element.

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

2.74.2 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
Footer

The following is the XML Schema definition of the **Footer.TableRows** element.

```
<xsd:element name="TableRows" type="TableRowsType" />
```

2.75 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
Header.FixedHeader

Child elements
Header.RepeatOnNewPage
Header.TableRows

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>
```

2.75.1 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
Header

The following is the XML Schema definition of the **Header.FixedHeader** element.

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

2.75.2 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
Header

The following is the XML Schema definition of the **Header.RepeatOnNewPage** element.

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

2.75.3 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
Header

The following is the XML Schema definition of the **Header.TableRows** element.

```
<xsd:element name="TableRows" type="TableRowsType" />
```

2.76 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
Table

Child elements
TableColumns.TableColumn

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

```
<xsd:complexType name="TableColumnsType">
  <xsd:sequence>
    <xsd:element name="TableColumn" type="TableColumnType"
                 maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

2.76.1 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
TableColumns

The following is the XML Schema definition of the **TableColumns.TableColumn** element.

```
<xsd:element name="TableColumn" type="TableColumnType" maxOccurs="unbounded" />
```

2.77 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
TableColumns

Child elements
TableColumn.FixedHeader
TableColumn.Visibility
TableColumn.Width

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>
```

2.77.1 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
TableColumn

The following is the XML Schema definition of the **TableColumn.FixedHeader** element.

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

2.77.2 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
TableColumn

The following is the XML Schema definition of the **TableColumn.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

2.77.3 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
TableColumn

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

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

2.78 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
Table

Child elements
TableGroups.TableGroup

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>
```

2.78.1 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
TableGroups

The following is the XML Schema definition of the **TableGroups.TableGroup** element.

```
<xsd:element name="TableGroup" type="TableGroupType" maxOccurs="unbounded" />
```

2.79 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
TableGroups

Child elements
TableGroup.Footer
TableGroup.Grouping
TableGroup.Header
TableGroup.Sorting
TableGroup.Visibility

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>
```

2.79.1 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" />
```

2.79.2 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" />
```

2.79.3 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" />
```

2.79.4 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" />
```

2.79.5 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" />
```

2.80 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
ChartMember
DataMember
GaugeMember
MapMember
TablixMember

Attributes
Group.Name

Child elements
Group.DataElementName
Group.DataElementOutput
Group.DocumentMapLabel
Group.DomainScope
Group.Filters
Group.GroupExpressions
Group.PageBreak
Group.PageName
Group.Parent
Group.ReGroupExpressions
Group.Variables

Applies to RDL 2011/01

Child elements
Group.DataSetName
Group.NaturalGroup
Group.Relationship

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:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
```

2.80.1 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" />
```

2.80.2 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" />
```

2.80.3 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>
```

2.80.4 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" />
```

2.80.5 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" />
```

2.80.6 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
Group

The following is the XML Schema definition of the **Group.DomainScope** element.

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

2.80.7 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
Group

The following is the XML Schema definition of the **Group.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

2.80.8 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" />
```

2.80.9 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" />
```

2.80.10 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" />
```

2.80.11 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
Group

The following is the XML Schema definition of the **Group.PageName** element.

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

2.80.12 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" />
```

2.80.13 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
Group

The following is the XML Schema definition of the **Group.ReGroupExpressions** element.

```
<xsd:element name="ReGroupExpressions" type="GroupExpressionsType" minOccurs="0" />
```

2.80.14 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" />
```

2.80.15 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" />
```

2.81 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
Group
Grouping

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>
```

2.81.1 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
GroupExpressions

The following is the XML Schema definition of the **GroupExpressions.GroupExpression** element.

```
<xsd:element name="GroupExpression" type="xsd:string" maxOccurs="unbounded" />
```

2.82 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
ChartMember
DataMember
GaugeMember
TablixMember
Chart
Tablix
GaugePanel
CustomData

Child elements
SortExpressions.SortExpression

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: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>
```

2.82.1 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.

Parent elements
SortExpressions

The following is the XML Schema definition of the **SortExpressions.SortExpression** element.

```
<xsd:element name="SortExpression" type="SortExpressionType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.83 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.

Parent elements
SortExpressions

Child elements
SortExpression.Direction
SortExpression.Value

Applies to RDL 2011/01

Child elements
SortExpression.NaturalSort
SortExpression.DeferredSort

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>
```

2.83.1 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>
```

2.83.2 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" />
```

2.83.3 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
SortExpression

The following is the XML Schema definition of the **SortExpression.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1" />
```

2.83.4 SortExpression.DeferredSort

Applies to RDL 2011/01

The **SortExpression.DeferredSort** element specifies whether an external component performs deferred sorting. The **SortExpression.DeferredfSort** 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
SortExpression

The following is the XML Schema definition of the **SortExpression.DeferredSort** element.

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

2.84 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
DataGrouping
DynamicColumns
DynamicRows
List
TableGroup
Details
DynamicCategories
DynamicSeries

Attributes
Grouping.Name

Child elements
Grouping.CustomProperties
Grouping.DataCollectionName
Grouping.DataElementName
Grouping.DataElementOutput
Grouping.Filters
Grouping.GroupExpressions
Grouping.Label
Grouping.PageBreakAtEnd
Grouping.PageBreakAtStart
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>
```

2.84.1 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" />
```

2.84.2 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
Grouping

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.

2.84.3 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" />
```

2.84.4 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" />
```

2.84.5 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>
```

2.84.6 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" />
```

2.84.7 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" />
```

2.84.8 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" />
```

2.84.9 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
Grouping

The following is the XML Schema definition of the **Grouping.PageBreakAtEnd** element.

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

2.84.10 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
Grouping

The following is the XML Schema definition of the **Grouping.PageBreakAtStart** element.

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

2.84.11 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
Grouping

The following is the XML Schema definition of the **Grouping.Parent** element.

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

2.85 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
DataGrouping
DynamicColumns
DynamicRows
List
TableGroup
Details
DynamicCategories
DynamicSeries

Child elements
Sorting.SortBy

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>
```

2.85.1 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
Sorting

The following is the XML Schema definition of the **Sorting.SortBy** element.

```
<xsd:element name="SortBy" type="SortByType" maxOccurs="unbounded" />
```

2.86 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
Sorting

Child elements
SortBy.Direction
SortBy.SortExpression

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>
```

2.86.1 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
SortBy

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>
```

2.86.2 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
SortBy

The following is the XML Schema definition of the **SortBy.SortExpression** element.

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

2.87 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
ReportItems
CustomReportItem.AltReportItem

The following are additional parent elements of the **Chart** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Parent elements
CellContents

The following are the attributes and child elements of the **Chart** element.

Attributes
Chart.Name

Child elements
Chart.Bookmark
Chart.ChartData
Chart.CustomProperties
Chart.DataElementName
Chart.DataElementOutput
Chart.DataSetName
Chart.Filters
Chart.Height
Chart.Left
Chart.Palette
Chart.Style
Chart.ToolTip
Chart.Top
Chart.Visibility
Chart.Width
Chart.ZIndex

The following are additional child elements of the **Chart** element in RDL 2003/10 and RDL 2005/01.

Child elements
Chart.Action
Chart.CategoryAxis
Chart.CategoryGroupings
Chart.ChartElementOutput
Chart.KeepTogether
Chart.Label
Chart.Legend
Chart.LinkToChild
Chart.PageBreakAtEnd
Chart.PageBreakAtStart
Chart.PlotArea
Chart.PointWidth
Chart.NoRows
Chart.SeriesGroupings
Chart.Subtype
Chart.ThreeDProperties
Chart.Title
Chart.Type
Chart.ValueAxis

The following are additional child elements of the **Chart** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Child elements
Chart.ActionInfo
Chart.DocumentMapLabel
Chart.RepeatWith
Chart.NoRowsMessage
Chart.PageBreak
Chart.SortExpressions
Chart.ChartAreas
Chart.ChartBorderSkin
Chart.ChartCategoryHierarchy
Chart.ChartCustomPaletteColors

Child elements
Chart.ChartLegends
Chart.ChartNoDataMessage
Chart.ChartSeriesHierarchy
Chart.ChartTitles
Chart.DynamicHeight
Chart.DynamicWidth
Chart.PaletteHatchBehavior

The following are additional child elements of the **Chart** element in RDL 2008/01.

Child elements
Chart.ChartAnnotations
Chart.ChartCodeParameters
Chart.Code
Chart.CodeLanguage

The following are additional child elements of the **Chart** element in RDL 2010/01, and RDL 2016/01.

Child elements
Chart.PageName

Applies to RDL 2011/01

Child elements
Chart.Relationship

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="ChartDataType" 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="ChartDataType" 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>
```

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="ChartDataType" 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>
```

2.87.1 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
Chart

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.87.2 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
Chart

The following is the XML Schema definition of the **Chart.Action** element.

```
<xsd:element name="Action" type="ActionType" minOccurs="0" />
```

2.87.3 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
Chart

The following is the XML Schema definition of the **Chart.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.87.4 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
Chart

The following is the XML Schema definition of the **Chart.Bookmark** element.

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

2.87.5 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" />
```

2.87.6 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" />
```

2.87.7 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
Chart

The following is the XML Schema definition of the **Chart.ChartAnnotations** element.

```
<xsd:element name="ChartAnnotations" type="ChartAnnotationsType" minOccurs="0" />
```

2.87.8 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
Chart

The following is the XML Schema definition of the **Chart.ChartAreas** element.

```
<xsd:element name="ChartAreas" type="ChartAreasType" minOccurs="0" />
```

2.87.9 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
Chart

The following is the XML Schema definition of the **Chart.ChartBorderSkin** element.

```
<xsd:element name="ChartBorderSkin" type="ChartBorderSkinType" minOccurs="0" />
```

2.87.10 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
Chart

The following is the XML Schema definition of the **Chart.ChartCodeParameters** element.

```
<xsd:element name="ChartCodeParameters" type="ChartCodeParametersType" minOccurs="0" />
```

2.87.11 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
Chart

The following is the XML Schema definition of the **Chart.ChartCategoryHierarchy** element.

```
<xsd:element name="ChartCategoryHierarchy" type="ChartHierarchyType" />
```

2.87.12 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
Chart

The following is the XML Schema definition of the **Chart.ChartCustomPaletteColors** element.

```
<xsd:element name="ChartCustomPaletteColors" type="ChartCustomPaletteColorsType"
             minOccurs="0" />
```

2.87.13 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
Chart

The following is the XML Schema definition of the **Chart.ChartData** element.

```
<xsd:element name="ChartData" type="ChartDataType" minOccurs="0" />
```

2.87.14 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
Chart

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>
```

2.87.15 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
Chart

The following is the XML Schema definition of the **Chart.ChartLegends** element.

```
<xsd:element name="ChartLegends" type="ChartLegendsType" minOccurs="0" />
```

2.87.16 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" />
```

2.87.17 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" />
```

2.87.18 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" />
```

2.87.19 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
Chart

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

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

2.87.20 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
Chart

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>
```

2.87.21 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
Chart

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.

2.87.22 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" />
```

2.87.23 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>
```

2.87.24 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" />
```

2.87.25 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" />
```

2.87.26 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" />
```

2.87.27 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" />
```

2.87.28 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" />
```

2.87.29 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
Chart

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

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

2.87.30 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
Chart

The following is the XML Schema definition of the **Chart.KeepTogether** element.

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

2.87.31 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
Chart

The following is the XML Schema definition of the **Chart.Label** element.

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

2.87.32 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 RdlSize. 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" />
```

2.87.33 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" />
```

2.87.34 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
Chart

The following is the XML Schema definition of the **Chart.LinkToChild** element.

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

2.87.35 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 ([XMLSCHEMA2/2] 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" />
```

2.87.36 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
Chart

The following is the XML Schema definition of the **Chart.NoRowsMessage** element.

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

2.87.37 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
Chart

The following is the XML Schema definition of the **Chart.PageBreak** element.

```
<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
```

2.87.38 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
Chart

The following is the XML Schema definition of the **Chart.PageBreakAtEnd** element.

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

2.87.39 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
Chart

The following is the XML Schema definition of the **Chart.PageBreakAtStart** element.

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

2.87.40 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
Chart

The following is the XML Schema definition of the **Chart.PageName** element.

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

2.87.41 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" />
```

2.87.42 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
Chart

The following is the XML Schema definition of the **Chart.PaletteHatchBehavior** element.

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

2.87.43 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
Chart

The following is the XML Schema definition of the **Chart.PlotArea** element.

```
<xsd:element name="PlotArea" type="PlotAreaType" minOccurs="0" />
```

2.87.44 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
Chart

The following is the XML Schema definition of the **Chart.PointWidth** element.

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

2.87.45 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
Chart

The following is the XML Schema definition of the **Chart.Relationship** element.

```
<xsd:element name ="Relationship" type="RelationshipType" minOccurs="0" />
```

2.87.46 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
Chart

The following is the XML Schema definition of the **Chart.RepeatWith** element.

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

2.87.47 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
Chart

The following is the XML Schema definition of the **Chart.SeriesGroupings** element.

```
<xsd:element name="SeriesGroupings" type="SeriesGroupingsType" minOccurs="0" />
```

2.87.48 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
Chart

The following is the XML Schema definition of the **Chart.SortExpressions** element.

```
<xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
```

2.87.49 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
Chart

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

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

2.87.50 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
Chart

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>
```

2.87.51 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
Chart

The following is the XML Schema definition of the **Chart.ThreeDProperties** element.

```
<xsd:element name="ThreeDProperties" type="ThreeDPropertiesType" minOccurs="0" />
```

2.87.52 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" />
```

2.87.53 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" />
```

2.87.54 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" />
```

2.87.55 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>
```

2.87.56 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" />
```

2.87.57 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" />
```

2.87.58 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" />
```

2.87.59 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
Chart

The following is the XML Schema definition of the **Chart.ZIndex** element.

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

2.88 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
Chart

Child elements
CategoryAxis.Axis

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>
```

2.88.1 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
CategoryAxis

The following is the XML Schema definition of the **CategoryAxis.Axis** element.

```
<xsd:element name="Axis" type="AxisType" minOccurs="0" />
```

2.89 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
CategoryAxis
ValueAxis

Child elements
Axis.CrossAt
Axis.Interlaced
Axis.LogScale
Axis.MajorGridLines
Axis.MajorInterval
Axis.MajorTickMarks
Axis.Margin
Axis.Max
Axis.Min
Axis.MinorGridLines
Axis.MinorInterval
Axis.MinorTickMarks
Axis.Reverse
Axis.Scalar
Axis.Style
Axis.Title
Axis.Visible

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: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>
```

2.89.1 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" />
```

2.89.2 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" />
```

2.89.3 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" />
```

2.89.4 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" />
```

2.89.5 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" />
```

2.89.6 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>
```

2.89.7 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
Axis

The following is the XML Schema definition of the **Axis.Margin** element.

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

2.89.8 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
Axis

The following is the XML Schema definition of the **Axis.Max** element.

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

2.89.9 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
Axis

The following is the XML Schema definition of the **Axis.Min** element.

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

2.89.10 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" />
```

2.89.11 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" />
```

2.89.12 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>
```

2.89.13 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" />
```

2.89.14 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" />
```

2.89.15 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
Axis

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

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

2.89.16 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" />
```

2.89.17 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
Axis

The following is the XML Schema definition of the **Axis.Visible** element.

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

2.90 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
Axis

Child elements
MajorGridLines.ShowGridLines
MajorGridLines.Style

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>
```

2.90.1 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
MajorGridLines

The following is the XML Schema definition of the **MajorGridLines.ShowGridLines** element.

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

2.90.2 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
MajorGridLines

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

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

2.91 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
Axis

Child elements
MinorGridLines.ShowGridLines
MinorGridLines.Style

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>
```

2.91.1 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
MinorGridLines

The following is the XML Schema definition of the **MinorGridLines.ShowGridLines** element.

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

2.91.2 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
MinorGridLines

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

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

2.92 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
CategoryGroupings.CategoryGrouping

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

```
<xsd:complexType name="CategoryGroupingsType">
  <xsd:sequence>
    <xsd:element name="CategoryGrouping" type="CategoryGroupingType"
```

```
                maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

2.92.1 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
CategoryGroupings

The following is the XML Schema definition of the **CategoryGroupings.CategoryGrouping** element.

```
<xsd:element name="CategoryGrouping" type="CategoryGroupingType"
             maxOccurs="unbounded" />
```

2.93 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
CategoryGroupings

Child elements
CategoryGrouping.DynamicCategories
CategoryGrouping.StaticCategories

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>
```

2.93.1 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
CategoryGrouping

The following is the XML Schema definition of the **CategoryGrouping.DynamicCategories** element.

```
<xsd:element name="DynamicCategories" type="DynamicCategoriesType"
             minOccurs="0" />
```

2.93.2 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
CategoryGrouping

The following is the XML Schema definition of the **CategoryGrouping.StaticCategories** element.

```
<xsd:element name="StaticCategories" type="StaticCategoriesType" minOccurs="0" />
```

2.94 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
CategoryGrouping

Child elements
DynamicCategories.Grouping
DynamicCategories.Label
DynamicCategories.Sorting

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>
```

2.94.1 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.

Parent elements
DynamicCategories

The following is the XML Schema definition of the **DynamicCategories.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" />
```

2.94.2 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.

Parent elements
DynamicCategories

The following is the XML Schema definition of the **DynamicCategories.Label** element.

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

2.94.3 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
DynamicCategories

The following is the XML Schema definition of the **DynamicCategories.Sorting** element.

```
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
```

2.95 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
CategoryGrouping

Child elements
StaticCategories.StaticMember

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

```
<xsd:complexType name="StaticCategoriesType">
  <xsd:sequence>
    <xsd:element name="StaticMember" type="StaticMemberType"
                 maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

2.95.1 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
StaticCategories

The following is the XML Schema definition of the **StaticCategories.StaticMember** element.

```
<xsd:element name="StaticMember" type="StaticMemberType" maxOccurs="unbounded" />
```

2.96 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
StaticCategories
StaticSeries

Child elements
StaticMember.Label

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>
```

2.96.1 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
StaticMember

The following is the XML Schema definition of the **StaticMember.Label** element.

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

2.97 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
Chart

The following are the child elements of the **ChartData** element in RDL 2003/10 and RDL 2005/01.

Child elements
ChartData.ChartSeries

The following are the child elements of the **ChartData** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Child elements
ChartData.ChartSeriesCollection
ChartData.ChartDerivedSeriesCollection

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

```
<xsd:complexType name="ChartDataType">
  <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="ChartDataType">
  <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="ChartDataType">
  <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>
```

2.97.1 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.

Parent elements
ChartData

The following is the XML Schema definition of the **ChartData.ChartSeries** element.

```
<xsd:element name="ChartSeries" type="ChartSeriesType" maxOccurs="unbounded" />
```

2.97.2 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.

Parent elements
ChartData

The following is the XML Schema definition of the **ChartData.ChartSeriesCollection** element.

```
<xsd:element name="ChartSeriesCollection" type="ChartSeriesCollectionType"
             minOccurs="1" maxOccurs="1" />
```

2.97.3 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.

Parent elements
ChartData

The following is the XML Schema definition of the **ChartData.ChartDerivedSeriesCollection** element.

```
<xsd:element name="ChartDerivedSeriesCollection"
             type="ChartDerivedSeriesCollectionType" minOccurs="0" />
```

2.98 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.

Parent elements
ChartData

The following are the parent elements of the **ChartSeries** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Parent elements
ChartSeriesCollection
ChartDerivedSeries

The following are the attributes of the **ChartSeries** element.

Attributes
ChartSeries.Name

The following are the child elements of the **ChartSeries** element in RDL 2003/10 and RDL 2005/01.

Child elements
ChartSeries.DataPoints
ChartSeries.PlotType

The following are the child elements of the **ChartSeries** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Child elements
ChartSeries.CategoryAxisName
ChartSeries.ChartAreaName
ChartSeries.ChartDataLabel
ChartSeries.ChartDataPoints
ChartSeries.ChartEmptyPoints
ChartSeries.ChartItemInLegend
ChartSeries.ChartMarker
ChartSeries.ChartSmartLabel
ChartSeries.CustomProperties
ChartSeries.Hidden
ChartSeries.LegendName
ChartSeries.Style
ChartSeries.Subtype
ChartSeries.Type
ChartSeries.ValueAxisName

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"
```

```
                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>
```

2.98.1 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
ChartSeries

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.98.2 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" />
```

2.98.3 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" />
```

2.98.4 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" />
```

2.98.5 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" />
```

2.98.6 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
ChartSeries

The following is the XML Schema definition of the **ChartSeries.ChartEmptyPoints** element.

```
<xsd:element name="ChartEmptyPoints" type="ChartEmptyPointsType" minOccurs="0" />
```

2.98.7 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" />
```

2.98.8 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" />
```

2.98.9 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" />
```

2.98.10 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" />
```

2.98.11 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" />
```

2.98.12 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" />
```

2.98.13 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" />
```

2.98.14 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>
```

2.98.15 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.

Parent elements
ChartSeries

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

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

2.98.16 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 ([XMLSCHEMA2/2] 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:

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" />
```

2.98.17 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" />
```

2.98.18 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" />
```

2.99 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.

Parent elements
ChartSeries

Child elements
DataPoints.DataPoint

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>
```

2.99.1 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.

Parent elements
DataPoints

The following is the XML Schema definition of the **DataPoints.DataPoint** element.

```
<xsd:element name="DataPoint" type="DataPointType" maxOccurs="unbounded" />
```

2.100 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.

Parent elements
DataPoints

Child elements
DataPoint.Action

Child elements
DataPoint.DataElementName
DataPoint.DataElementOutput
DataPoint.DataLabel
DataPoint.DataValues
DataPoint.Marker
DataPoint.Style

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>
```

2.100.1 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" />
```

2.100.2 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" />
```

2.100.3 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>
```

2.100.4 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" />
```

2.100.5 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" />
```

2.100.6 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
DataPoint

The following is the XML Schema definition of the **DataPoint.Marker** element.

```
<xsd:element name="Marker" type="MarkerType" minOccurs="0" />
```

2.100.7 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" />
```

2.101 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
DataPoint

Child elements
DataLabel.Position
DataLabel.Rotation
DataLabel.Style
DataLabel.Value
DataLabel.Visible

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" />
```

```
</xsd:complexType>
```

2.101.1 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 ([XMLSCHEMA2/2] 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>
```

2.101.2 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
DataLabel

The following is the XML Schema definition of the **DataLabel.Rotation** element.

```
<xsd:element name="Rotation" type="xsd:integer" minOccurs="0" />
```

2.101.3 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
DataLabel

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

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

2.101.4 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
DataLabel

The following is the XML Schema definition of the **DataLabel.Value** element.

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

2.101.5 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
DataLabel

The following is the XML Schema definition of the **DataLabel.Visible** element.

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

2.102 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
DataPoint

Child elements
DataValues.DataValue

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>
```

2.102.1 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.

Parent elements
DataValues

The following is the XML Schema definition of the **DataValues.DataValue** element.

```
<xsd:element name="DataValue" type="DataValueType" minOccurs="unbounded" />
```

2.103 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.

Parent elements
DataPoint

Child elements
Marker.Size
Marker.Style
Marker.Type

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>
```

2.103.1 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 RdlSize or an expression that evaluates to an **RdlSize**. If this element is not present, the marker will not appear.

Following is the parent element of the **Marker.Size** element.

Parent elements
Marker

The following is the XML Schema definition of the **Marker.Size** element.

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

2.103.2 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
Marker

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

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

2.103.3 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.

Parent elements
Marker

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>
```

2.104 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.

Parent elements
Chart

Child elements
Legend.InsidePlotArea
Legend.Layout
Legend.Position
Legend.Visible

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: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>
```

2.104.1 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
Legend

The following is the XML Schema definition of the **Legend.InsidePlotArea** element.

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

2.104.2 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.

Parent elements
Legend

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>
```

2.104.3 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.

Parent elements
Legend

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>
```

2.104.4 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
Legend

The following is the XML Schema definition of the **Legend.Visible** element.

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

2.105 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
Chart

Child elements
PlotArea.Style

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>
```

2.105.1 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
PlotArea

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

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

2.106 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
SeriesGroupings.SeriesGrouping

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

```
<xsd:complexType name="SeriesGroupingsType">
  <xsd:sequence>
    <xsd:element name="SeriesGrouping" type="SeriesGroupingType"
                 maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

2.106.1 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
SeriesGroupings

The following is the XML Schema definition of the **SeriesGroupings.SeriesGrouping** element.

```
<xsd:element name="SeriesGrouping" type="SeriesGroupingType"
             maxOccurs="unbounded" />
```

2.107 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
SeriesGroupings

Child elements
SeriesGrouping.DynamicSeries
SeriesGrouping.StaticSeries

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>
```

2.107.1 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.

Parent elements
SeriesGrouping

The following is the XML Schema definition of the **SeriesGrouping.DynamicSeries** element.

```
<xsd:element name="DynamicSeries" type="DynamicSeriesType" minOccurs="0" />
```

2.107.2 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.

Parent elements
SeriesGrouping

The following is the XML Schema definition of the **SeriesGrouping.StaticSeries** element.

```
<xsd:element name="StaticSeries" type="StaticSeriesType" minOccurs="0" />
```

2.108 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.

Parent elements
SeriesGrouping

Child elements
DynamicSeries.Grouping
DynamicSeries.Label
DynamicSeries.Sorting

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

2.108.1 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
DynamicSeries

The following is the XML Schema definition of the **DynamicSeries.Grouping** element.

```
<xsd:element name="Grouping" type="GroupingType" />
```

2.108.2 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
DynamicSeries

The following is the XML Schema definition of the **DynamicSeries.Label** element.

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

2.108.3 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
DynamicSeries

The following is the XML Schema definition of the **DynamicSeries.Sorting** element.

```
<xsd:element name="Sorting" type="SortingType" minOccurs="0" />
```

2.109 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
SeriesGrouping

Child elements
StaticSeries.StaticMember

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

```
<xsd:complexType name="StaticSeriesType">
  <xsd:sequence>
    <xsd:element name="StaticMember" type="StaticMemberType"
                 maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

2.109.1 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
StaticSeries

The following is the XML Schema definition of the **StaticSeries.StaticMember** element.

```
<xsd:element name="StaticMember" type="StaticMemberType" maxOccurs="unbounded" />
```

2.110 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
ThreeDProperties.Clustered
ThreeDProperties.DepthRatio
ThreeDProperties.DrawingStyle
ThreeDProperties.Enabled
ThreeDProperties.GapDepth
ThreeDProperties.HeightRatio
ThreeDProperties.Inclination
ThreeDProperties.Perspective
ThreeDProperties.ProjectionMode
ThreeDProperties.Rotation
ThreeDProperties.Shading
ThreeDProperties.WallThickness

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>
```

2.110.1 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.

Parent elements
ThreeDProperties

The following is the XML Schema definition of the **ThreeDProperties.Clustered** element.

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

2.110.2 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.

Parent elements
ThreeDProperties

The following is the XML Schema definition of the **ThreeDProperties.DepthRatio** element.

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

2.110.3 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.

Parent elements
ThreeDProperties

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>
```

2.110.4 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.

Parent elements
ThreeDProperties

The following is the XML Schema definition of the **ThreeDProperties.Enabled** element.

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

2.110.5 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.

Parent elements
ThreeDProperties

The following is the XML Schema definition of the **ThreeDProperties.GapDepth** element.

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

2.110.6 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
ThreeDProperties

The following is the XML Schema definition of the **ThreeDProperties.HeightRatio** element.

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

2.110.7 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
ThreeDProperties

The following is the XML Schema definition of the **ThreeDProperties.Inclination** element.

```
<xsd:element name="Inclination" type="xsd:integer" minOccurs="0" />
```

2.110.8 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
ThreeDProperties

The following is the XML Schema definition of the **ThreeDProperties.Perspective** element.

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

2.110.9 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.

Ortographic: Specifies that ortographic 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
ThreeDProperties

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>
```

2.110.10 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.

Parent elements
ThreeDProperties

The following is the XML Schema definition of the **ThreeDProperties.Rotation** element.

```
<xsd:element name="Rotation" type="xsd:integer" minOccurs="0" />
```

2.110.11 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.

Parentelements
ThreeDProperties

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>
```

2.110.12 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.

Parent elements
ThreeDProperties

The following is the XML Schema definition of the **ThreeDProperties.WallThickness** element.

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

2.111 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.

Parent elements
Chart
Axis

Child elements
Title.Caption
Title.Position
Title.Style

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>
```

2.111.1 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
Title

The following is the XML Schema definition of the **Title.Caption** element.

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

2.111.2 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
Title

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>
```

2.111.3 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.

Parent elements
Title

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

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

2.112 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.

Parent elements
Chart

Child elements
ValueAxis.Axis

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>
```

2.112.1 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.

Parent elements
ValueAxis

The following is the XML Schema definition of the **ValueAxis.Axis** element.

```
<xsd:element name="Axis" type="AxisType" minOccurs="0" />
```

2.113 ChartAnnotations

Applies to RDL 2008/01

The **ChartAnnotations** element is ignored.

Following are the parent and child elements of the **ChartAnnotations** element.

Parent elements
Chart

Child elements
ChartAnnotations.ChartAnnotation

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>
```

2.113.1 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
ChartAnnotations

The following is the XML Schema definition of the **ChartAnnotations.ChartAnnotation** element.

```
<xsd:element name="ChartAnnotation" type="ChartAnnotationType" maxOccurs="unbounded" />
```

2.114 ChartAnnotation

Applies to RDL 2008/01

The **ChartAnnotation** element is ignored.

Following is the parent element of the **ChartAnnotation** element.

Parent elements
ChartAnnotations

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>
```

2.115 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
ChartAreas.ChartArea

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>
```

2.115.1 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" />
```

2.116 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
ChartAreas

Attributes
ChartArea.Name

Child elements
ChartArea.AlignOrientation
ChartArea.AlignWithChartArea
ChartArea.ChartAlignType
ChartArea.ChartCategoryAxes
ChartArea.ChartElementPosition
ChartArea.ChartInnerPlotPosition
ChartArea.ChartThreeDProperties
ChartArea.ChartValueAxes
ChartArea.EquallySizedAxesFont
ChartArea.Hidden
ChartArea.Style

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: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>
```

2.116.1 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" />
```

2.116.2 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" />
```

2.116.3 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" />
```

2.116.4 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
ChartArea

The following is the XML Schema definition of the **ChartArea.ChartAlignType** element.

```
<xsd:element name="ChartAlignType" type="ChartAlignTypeType" minOccurs="0" />
```

2.116.5 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
ChartArea

The following is the XML Schema definition of the **ChartArea.ChartCategoryAxes** element.

```
<xsd:element name="ChartCategoryAxes" type="ChartCategoryAxesType" minOccurs="0" />
```

2.116.6 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
ChartArea

The following is the XML Schema definition of the **ChartArea.ChartElementPosition** element.

```
<xsd:element name="ChartElementPosition" type="ChartElementPositionType"
             minOccurs="0" />
```

2.116.7 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
ChartArea

The following is the XML Schema definition of the **ChartArea.ChartInnerPlotPosition** element.

```
<xsd:element name="ChartInnerPlotPosition" type="ChartElementPositionType"
             minOccurs="0" />
```

2.116.8 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
ChartArea

The following is the XML Schema definition of the **ChartArea.ChartThreeDProperties** element.

```
<xsd:element name="ChartThreeDProperties" type="ChartThreeDPropertiesType"
             minOccurs="0" />
```

2.116.9 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
ChartArea

The following is the XML Schema definition of the **ChartArea.ChartValueAxes** element.

```
<xsd:element name="ChartValueAxes" type="ChartValueAxesType" minOccurs="0" />
```

2.116.10 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
ChartArea

The following is the XML Schema definition of the **ChartArea.EquallySizedAxesFont** element.

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

2.116.11 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
ChartArea

The following is the XML Schema definition of the **ChartArea.Hidden** element.

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

2.116.12 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
ChartArea

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

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

2.117 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
ChartArea

Child elements
ChartAlignType.AxesView
ChartAlignType.Cursor
ChartAlignType.InnerPlotPosition
ChartAlignType.Position

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>
```

2.117.1 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
ChartAlignType

The following is the XML Schema definition of the **ChartAlignType.AxesView** element.

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

2.117.2 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
ChartAlignType

The following is the XML Schema definition of the **ChartAlignType.Cursor** element.

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

2.117.3 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
ChartAlignType

The following is the XML Schema definition of the **ChartAlignType.InnerPlotPosition** element.

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

2.117.4 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
ChartAlignType

The following is the XML Schema definition of the **ChartAlignType.Position** element.

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

2.118 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
ChartArea

Child elements
ChartCategoryAxes.ChartAxis

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>
```

2.118.1 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
ChartCategoryAxes

The following is the XML Schema definition of the **ChartCategoryAxes.ChartAxis** element.

```
<xsd:element name="ChartAxis" type="ChartAxisType" maxOccurs="unbounded" />
```

2.119 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
ChartCategoryAxes
ChartValueAxes

Attributes
ChartAxis.Name

Child elements
ChartAxis.AllowLabelRotation
ChartAxis.Angle
ChartAxis.Arrows
ChartAxis.ChartAxisScaleBreak
ChartAxis.ChartAxisTitle

Child elements
ChartAxis.ChartMajorGridLines
ChartAxis.ChartMajorTickMarks
ChartAxis.ChartMinorGridLines
ChartAxis.ChartMinorTickMarks
ChartAxis.ChartStripLines
ChartAxis.CrossAt
ChartAxis.CustomProperties
ChartAxis.HideEndLabels
ChartAxis.HideLabels
ChartAxis.IncludeZero
ChartAxis.Interlaced
ChartAxis.InterlacedColor
ChartAxis.Interval
ChartAxis.IntervalOffset
ChartAxis.IntervalOffsetType
ChartAxis.IntervalType
ChartAxis.LabelInterval
ChartAxis.LabelIntervalOffset
ChartAxis.LabelIntervalOffsetType
ChartAxis.LabelIntervalType
ChartAxis.LabelsAutoFitDisabled
ChartAxis.Location
ChartAxis.LogBase
ChartAxis.LogScale
ChartAxis.Margin
ChartAxis.MarksAlwaysAtPlotEdge
ChartAxis.MaxFontSize
ChartAxis.Maximum
ChartAxis.MinFontSize
ChartAxis.Minimum
ChartAxis.OffsetLabels
ChartAxis.PreventFontGrow

Child elements
ChartAxis.PreventFontShrink
ChartAxis.PreventLabelOffset
ChartAxis.PreventWordWrap
ChartAxis.Reverse
ChartAxis.Scalar
ChartAxis.Style
ChartAxis.VariableAutoInterval
ChartAxis.Visible

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: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>
```

2.119.1 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
ChartAxis

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.119.2 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.AllowLabelRotation** element.

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

2.119.3 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.Angle** element.

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

2.119.4 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.Arrows** element.

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

2.119.5 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.ChartAxisScaleBreak** element.

```
<xsd:element name="ChartAxisScaleBreak" type="ChartAxisScaleBreakType"
             minOccurs="0" />
```

2.119.6 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" />
```

2.119.7 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" />
```

2.119.8 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" />
```

2.119.9 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" />
```

2.119.10 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" />
```

2.119.11 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" />
```

2.119.12 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.CrossAt** element.

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

2.119.13 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
ChartAxis

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

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

2.119.14 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.HideEndLabels** element.

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

2.119.15 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.HideLabels** element.

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

2.119.16 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.IncludeZero** element.

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

2.119.17 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.Interlaced** element.

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

2.119.18 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" />
```

2.119.19 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" />
```

2.119.20 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.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.IntervalOffset** element.

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

2.119.21 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.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.IntervalOffsetType** element.

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

2.119.22 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.IntervalType** element.

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

2.119.23 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.LabelInterval** element.

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

2.119.24 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.LabelIntervalOffset** element.

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

2.119.25 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.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.LabelIntervalOffsetType** element.

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

2.119.26 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.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.LabelIntervalType** element.

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

2.119.27 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.LabelsAutoFitDisabled** element.

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

2.119.28 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.Location** element.

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

2.119.29 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.LogBase** element.

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

2.119.30 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.LogScale** element.

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

2.119.31 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.Margin** element.

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

2.119.32 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.MarksAlwaysAtPlotEdge** element.

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

2.119.33 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 RdlSize or an expression that evaluates to an **RdlSize**. If this element is not present, its value is interpreted as 10pt.

Following is the parent element of the **ChartAxis.MaxFontSize** element.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.MaxFontSize** element.

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

2.119.34 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.Maximum** element.

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

2.119.35 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.MinFontSize** element.

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

2.119.36 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.Minimum** element.

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

2.119.37 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.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.OffsetLabels** element.

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

2.119.38 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.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.PreventFontGrow** element.

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

2.119.39 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.

Parent elements
ChartAxis

The following is the XML Schema definition of the **ChartAxis.PreventFontShrink** element.

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

2.119.40 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.PreventLabelOffset** element.

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

2.119.41 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.PreventWordWrap** element.

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

2.119.42 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.Reverse** element.

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

2.119.43 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.Scalar** element.

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

2.119.44 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
ChartAxis

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

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

2.119.45 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.VariableAutoInterval** element.

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

2.119.46 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
ChartAxis

The following is the XML Schema definition of the **ChartAxis.Visible** element.

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

2.120 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
ChartAxis

Child elements
ChartAxisScaleBreak.BreakLineType
ChartAxisScaleBreak.CollapsibleSpaceThreshold

Child elements
ChartAxisScaleBreak.Enabled
ChartAxisScaleBreak.IncludeZero
ChartAxisScaleBreak.MaxNumberOfBreaks
ChartAxisScaleBreak.Spacing
ChartAxisScaleBreak.Style

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>
```

2.120.1 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
ChartAxisScaleBreak

The following is the XML Schema definition of the **ChartAxisScaleBreak.BreakLineType** element.

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

2.120.2 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
ChartAxisScaleBreak

The following is the XML Schema definition of the **ChartAxisScaleBreak.CollapsibleSpaceThreshold** element.

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

2.120.3 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
ChartAxisScaleBreak

The following is the XML Schema definition of the **ChartAxisScaleBreak.Enabled** element.

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

2.120.4 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
ChartAxisScaleBreak

The following is the XML Schema definition of the **ChartAxisScaleBreak.IncludeZero** element.

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

2.120.5 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
ChartAxisScaleBreak

The following is the XML Schema definition of the **ChartAxisScaleBreak.MaxNumberOfBreaks** element.

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

2.120.6 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
ChartAxisScaleBreak

The following is the XML Schema definition of the **ChartAxisScaleBreak.Spacing** element.

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

2.120.7 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
ChartAxisScaleBreak

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

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

2.121 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
ChartAxisTitle.Caption
ChartAxisTitle.Position
ChartAxisTitle.Style
ChartAxisTitle.TextOrientation

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>
```

2.121.1 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.

Parent elements
ChartAxisTitle

The following is the XML Schema definition of the **ChartAxisTitle.Caption** element.

```
<xsd:element name="Caption" type="StringLocIDType" />
```

2.121.2 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
ChartAxisTitle

The following is the XML Schema definition of the **ChartAxisTitle.Position** element.

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

2.121.3 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
ChartAxisTitle

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

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

2.121.4 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
ChartAxisTitle

The following is the XML Schema definition of the **ChartAxisTitle.TextOrientation** element.

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

2.122 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
ChartAxis

Child elements
ChartGridLines.Enabled
ChartGridLines.Interval
ChartGridLines.IntervalOffset
ChartGridLines.IntervalOffsetType
ChartGridLines.IntervalType
ChartGridLines.Style

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: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>
```

2.122.1 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
ChartGridLines

The following is the XML Schema definition of the **ChartGridLines.Enabled** element.

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

2.122.2 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.

Parent elements
ChartGridLines

The following is the XML Schema definition of the **ChartGridLines.Interval** element.

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

2.122.3 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.

Parent elements
ChartGridLines

The following is the XML Schema definition of the **ChartGridLines.IntervalOffset** element.

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

2.122.4 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
ChartGridLines

The following is the XML Schema definition of the **ChartGridLines.IntervalOffsetType** element.

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

2.122.5 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
ChartGridLines

The following is the XML Schema definition of the **ChartGridLines.IntervalType** element.

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

2.122.6 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
ChartGridLines

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

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

2.123 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
ChartStripLines.ChartStripLine

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"
                 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>
```

2.123.1 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
ChartStripLines

The following is the XML Schema definition of the **ChartStripLines.ChartStripLine** element.

```
<xsd:element name="ChartStripLine" type="ChartStripLineType"
             maxOccurs="unbounded" />
```

2.124 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
ChartStripLines

Child elements
ChartStripLine.ActionInfo
ChartStripLine.Interval
ChartStripLine.IntervalOffset
ChartStripLine.IntervalOffsetType
ChartStripLine.IntervalType
ChartStripLine.StripWidth
ChartStripLine.StripWidthType
ChartStripLine.Style
ChartStripLine.TextOrientation

Child elements
ChartStripLine.Title
ChartStripLine.TitleAngle
ChartStripLine.ToolTip

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>
```

2.124.1 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.

Parent elements
ChartStripLine

The following is the XML Schema definition of the **ChartStripLine.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.124.2 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.

Parent elements
ChartStripLine

The following is the XML Schema definition of the **ChartStripLine.Interval** element.

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

2.124.3 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.

Parent elements
ChartStripLine

The following is the XML Schema definition of the **ChartStripLine.IntervalOffset** element.

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

2.124.4 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
ChartStripLine

The following is the XML Schema definition of the **ChartStripLine.IntervalOffsetType** element.

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

2.124.5 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
ChartStripLine

The following is the XML Schema definition of the **ChartStripLine.IntervalType** element.

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

2.124.6 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
ChartStripLine

The following is the XML Schema definition of the **ChartStripLine.StripWidth** element.

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

2.124.7 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
ChartStripLine

The following is the XML Schema definition of the **ChartStripLine.StripWidthType** element.

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

2.124.8 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
ChartStripLine

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

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

2.124.9 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
ChartStripLine

The following is the XML Schema definition of the **ChartStripLine.TextOrientation** element.

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

2.124.10 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
ChartStripLine

The following is the XML Schema definition of the **ChartStripLine.Title** element.

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

2.124.11 ChartStripLine.TitleAngle

Applies to RDL 2008/01

The **ChartStripLine.TitleAngle** element is ignored.

Following is the parent element of the **ChartStripLine.TitleAngle** element.

Parent elements
ChartStripLine

The following is the XML Schema definition of the **ChartStripLine.TitleAngle** element.

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

2.124.12 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.

Parent elements
ChartStripLine

The following is the XML Schema definition of the **ChartStripLine.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

2.125 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.

Parent elements
ChartAxis

Child elements
ChartTickMarks.Enabled
ChartTickMarks.Interval
ChartTickMarks.IntervalOffset
ChartTickMarks.IntervalOffsetType
ChartTickMarks.IntervalType
ChartTickMarks.Length

Child elements
ChartTickMarks.Style
ChartTickMarks.Type

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>
```

2.125.1 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.

Parent elements
ChartTickMarks

The following is the XML Schema definition of the **ChartTickMarks.Enabled** element.

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

2.125.2 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.

Parent elements
ChartTickMarks

The following is the XML Schema definition of the **ChartTickMarks.Interval** element.

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

2.125.3 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.

Parent elements
ChartTickMarks

The following is the XML Schema definition of the **ChartTickMarks.IntervalOffset** element.

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

2.125.4 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.

Parent elements
ChartTickMarks

The following is the XML Schema definition of the **ChartTickMarks.IntervalOffsetType** element.

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

2.125.5 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
ChartTickMarks

The following is the XML Schema definition of the **ChartTickMarks.IntervalType** element.

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

2.125.6 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
ChartTickMarks

The following is the XML Schema definition of the **ChartTickMarks.Length** element.

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

2.125.7 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.

Parent elements
ChartTickMarks

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

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

2.125.8 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.

Parent elements
ChartTickMarks

The following is the XML Schema definition of the **ChartTickMarks.Type** element.

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

2.126 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.

Parent elements
ChartArea
ChartLegend
ChartTitle

Child elements
ChartElementPosition.Height
ChartElementPosition.Left
ChartElementPosition.Top
ChartElementPosition.Width

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>
```

2.126.1 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
ChartElementPosition

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

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

2.126.2 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
ChartElementPosition

The following is the XML Schema definition of the **ChartElementPosition.Left** element.

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

2.126.3 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
ChartElementPosition

The following is the XML Schema definition of the **ChartElementPosition.Top** element.

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

2.126.4 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
ChartElementPosition

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

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

2.127 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
ChartArea

Child elements
ChartThreeDProperties.Clustered
ChartThreeDProperties.DepthRatio
ChartThreeDProperties.Enabled
ChartThreeDProperties.GapDepth
ChartThreeDProperties.Inclination
ChartThreeDProperties.Perspective
ChartThreeDProperties.ProjectionMode
ChartThreeDProperties.Rotation
ChartThreeDProperties.Shading
ChartThreeDProperties.WallThickness

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: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>
```

2.127.1 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
ChartThreeDProperties

The following is the XML Schema definition of the **ChartThreeDProperties.Clustered** element.

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

2.127.2 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
ChartThreeDProperties

The following is the XML Schema definition of the **ChartThreeDProperties.DepthRatio** element.

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

2.127.3 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
ChartThreeDProperties

The following is the XML Schema definition of the **ChartThreeDProperties.Enabled** element.

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

2.127.4 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
ChartThreeDProperties

The following is the XML Schema definition of the **ChartThreeDProperties.GapDepth** element.

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

2.127.5 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
ChartThreeDProperties

The following is the XML Schema definition of the **ChartThreeDProperties.Inclination** element.

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

2.127.6 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
ChartThreeDProperties

The following is the XML Schema definition of the **ChartThreeDProperties.Perspective** element.

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

2.127.7 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
ChartThreeDProperties

The following is the XML Schema definition of the **ChartThreeDProperties.ProjectionMode** element.

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

2.127.8 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
ChartThreeDProperties

The following is the XML Schema definition of the **ChartThreeDProperties.Rotation** element.

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

2.127.9 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
ChartThreeDProperties

The following is the XML Schema definition of the **ChartThreeDProperties.Shading** element.

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

2.127.10 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
ChartThreeDProperties

The following is the XML Schema definition of the **ChartThreeDProperties.WallThickness** element.

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

2.128 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

Child elements
ChartValueAxes.ChartAxis

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>
```

2.128.1 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.

Parent elements
ChartValueAxes

The following is the XML Schema definition of the **ChartValueAxes.ChartAxis** element.

```
<xsd:element name="ChartAxis" type="ChartAxisType" minOccurs="1" />
```

2.129 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.

Parent elements
Chart

Child elements
ChartBorderSkin.ChartBorderSkinType
ChartBorderSkin.Style

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>
```

2.129.1 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
ChartBorderSkin

The following is the XML Schema definition of the **ChartBorderSkin.ChartBorderSkinType** element.

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

2.129.2 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
ChartBorderSkin

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

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

2.130 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
Chart

Child elements
ChartCodeParameters.ChartCodeParameter

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>
```

2.130.1 ChartCodeParameters.ChartCodeParameter

Applies to RDL 2008/01

The **ChartCodeParameters.ChartCodeParameter** element is ignored.

Following is the parent element of the **ChartCodeParameters.ChartCodeParameter** element.

Parent elements
ChartCodeParameters

The following is the XML Schema definition of the **ChartCodeParameters.ChartCodeParameter** element.

```
<xsd:element name="ChartCodeParameter" type="ChartCodeParameterType"
             maxOccurs="unbounded" />
```

2.131 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.

Parent elements
ChartCodeParameters

Attributes
ChartCodeParameter.Name

Child elements
ChartCodeParameter.Value

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>
```

2.131.1 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.

Parent elements
ChartCodeParameter

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.131.2 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.

Parent elements
ChartCodeParameter

The following is the XML Schema definition of the **ChartCodeParameter.Value** element.

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

2.132 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
Chart

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>
```

2.132.1 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
ChartCustomPaletteColors

The following is the XML Schema definition of the **ChartCustomPaletteColors.ChartCustomPaletteColor** element.

```
<xsd:element name="ChartCustomPaletteColor" type="xsd:string" maxOccurs="unbounded" />
```

2.133 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
ChartData

Child elements
ChartDerivedSeriesCollection.ChartDerivedSeries

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>
```

2.133.1 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" />
```

2.134 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
ChartDerivedSeriesCollection

Child elements
ChartDerivedSeries.ChartFormulaParameters
ChartDerivedSeries.ChartSeries
ChartDerivedSeries.DerivedSeriesFormula
ChartDerivedSeries.SourceChartSeriesName

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: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:element name="ChartFormulaParameters" type="ChartFormulaParametersType"
                 minOccurs="0" />
    <xsd:any namespace="##other" processContents="lax" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

2.134.1 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
ChartDerivedSeries

The following is the XML Schema definition of the **ChartDerivedSeries.ChartFormulaParameters** element.

```
<xsd:element name="ChartFormulaParameters" type="ChartFormulaParametersType"
             minOccurs="0" />
```

2.134.2 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
ChartDerivedSeries

The following is the XML Schema definition of the **ChartDerivedSeries.ChartSeries** element.

```
<xsd:element name="ChartSeries" type="ChartSeriesType" minOccurs="1" />
```

2.134.3 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.

Parent elements
ChartDerivedSeries

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: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: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: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>
```

2.134.4 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
ChartDerivedSeries

The following is the XML Schema definition of the **ChartDerivedSeries.SourceChartSeriesName** element.

```
<xsd:element name="SourceChartSeriesName" type="xsd:string" minOccurs="1" />
```

2.135 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
ChartDerivedSeries

Child elements
ChartFormulaParameters.ChartFormulaParameter

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>
```

2.135.1 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
ChartFormulaParameters

The following is the XML Schema definition of the **ChartFormulaParameters.ChartFormulaParameter** element.

```
<xsd:element name="ChartFormulaParameter" type="ChartFormulaParameterType"
             maxOccurs="unbounded" />
```

2.136 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
ChartFormulaParameters

Attributes
ChartFormulaParameter.Name

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>
```

2.136.1 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" />
```

2.136.2 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
ChartFormulaParameter

The following is the XML Schema definition of the **ChartFormulaParameter.Source** element.

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

2.136.3 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
ChartFormulaParameter

The following is the XML Schema definition of the **ChartFormulaParameter.Value** element.

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

2.137 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
ChartData

Child elements
ChartSeriesCollection.ChartSeries

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>
```

2.137.1 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
ChartSeriesCollection

The following is the XML Schema definition of the **ChartSeriesCollection.ChartSeries** element.

```
<xsd:element name="ChartSeries" type="ChartSeriesType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.138 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
ChartDataPoint
ChartEmptyPoints
ChartSeries

Child elements
ChartDataLabel.ActionInfo
ChartDataLabel.Label
ChartDataLabel.Position
ChartDataLabel.Rotation
ChartDataLabel.Style
ChartDataLabel.ToolTip
ChartDataLabel.UseValueAsLabel
ChartDataLabel.Visible

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>
```

2.138.1 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
ChartDataLabel

The following is the XML Schema definition of the **ChartDataLabel.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.138.2 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
ChartDataLabel

The following is the XML Schema definition of the **ChartDataLabel.Label** element.

```
<xsd:element name="Label" type="StringLocIDType" minOccurs="0" />
```

2.138.3 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
ChartDataLabel

The following is the XML Schema definition of the **ChartDataLabel.Position** element.

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

2.138.4 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
ChartDataLabel

The following is the XML Schema definition of the **ChartDataLabel.Rotation** element.

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

2.138.5 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
ChartDataLabel

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

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

2.138.6 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
ChartDataLabel

The following is the XML Schema definition of the **ChartDataLabel.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

2.138.7 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
ChartDataLabel

The following is the XML Schema definition of the **ChartDataLabel.UseValueAsLabel** element.

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

2.138.8 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.

Parent elements
ChartDataLabel

The following is the XML Schema definition of the **ChartDataLabel.Visible** element.

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

2.139 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.

Parent elements
ChartSeries

Child elements
ChartDataPoints.ChartDataPoint

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>
```

2.139.1 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
ChartDataPoints

The following is the XML Schema definition of the **ChartDataPoints.ChartDataPoint** element.

```
<xsd:element name="ChartDataPoint" type="ChartDataPointType" maxOccurs="unbounded" />
```

2.140 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
ChartDataPoints

Child elements
ChartDataPoint.ActionInfo
ChartDataPoint.AxisLabel
ChartDataPoint.ChartDataLabel
ChartDataPoint.ChartDataPointValues
ChartDataPoint.ChartItemInLegend
ChartDataPoint.ChartMarker
ChartDataPoint.CustomProperties
ChartDataPoint.DataElementName
ChartDataPoint.DataElementOutput
ChartDataPoint.Style
ChartDataPoint.ToolTip

Applies to RDL 2011/01

Child elements
ChartDataPoint.DataSetName
ChartDataPoint.Relationships

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>
```

2.140.1 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
ChartDataPoint

The following is the XML Schema definition of the **ChartDataPoint.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.140.2 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
ChartDataPoint

The following is the XML Schema definition of the **ChartDataPoint.AxisLabel** element.

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

2.140.3 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
ChartDataPoint

The following is the XML Schema definition of the **ChartDataPoint.ChartDataLabel** element.

```
<xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
```

2.140.4 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
ChartDataPoint

The following is the XML Schema definition of the **ChartDataPoint.ChartDataPointValues** element.

```
<xsd:element name="ChartDataPointValues" type="ChartDataPointValuesType"
             minOccurs="0" />
```

2.140.5 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
ChartDataPoint

The following is the XML Schema definition of the **ChartDataPoint.ChartItemInLegend** element.

```
<xsd:element name="ChartItemInLegend" type="ChartItemInLegendType"
             minOccurs="0" />
```

2.140.6 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
ChartDataPoint

The following is the XML Schema definition of the **ChartDataPoint.ChartMarker** element.

```
<xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
```

2.140.7 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
ChartDataPoint

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

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

2.140.8 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
ChartDataPoint

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

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

2.140.9 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.

Parent elements
ChartDataPoint

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>
```

2.140.10 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.

Parent elements
ChartDataPoint

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

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

2.140.11 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.

Parent elements
ChartDataPoint

The following is the XML Schema definition of the **ChartDataPoint.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

2.140.12 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" />
```

2.140.13 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 DataSetfor 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" />
```

2.141 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.

Parent elements
ChartDataPoint

Child elements
ChartDataPointValues.End
ChartDataPointValues.High
ChartDataPointValues.Low
ChartDataPointValues.Mean
ChartDataPointValues.Median
ChartDataPointValues.Size
ChartDataPointValues.Start
ChartDataPointValues.X
ChartDataPointValues.Y

Applies to RDL 2011/01

Child elements
ChartDataPointValues.HighlightX
ChartDataPointValues.HighlightY
ChartDataPointValues.HighlightSize

Applies to RDL 2013/01

Child elements
ChartDataPointValues.FormatX
ChartDataPointValues.FormatY
ChartDataPointValues.FormatSize
ChartDataPointValues.CurrencyLanguageX
ChartDataPointValues.CurrencyLanguageY
ChartDataPointValues.CurrencyLanguageSize

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: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>
```

2.141.1 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.

Parent elements
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.End** element.

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

2.141.2 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.High** element.

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

2.141.3 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.Low** element.

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

2.141.4 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.Mean** element.

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

2.141.5 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.Median** element.

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

2.141.6 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.Size** element.

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

2.141.7 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.Start** element.

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

2.141.8 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.X** element.

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

2.141.9 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.Y** element.

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

2.141.10 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.HighlightX** element.

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

2.141.11 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.HighlightY** element.

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

2.141.12 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.HighlightSize** element.

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

2.141.13 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.FormatX** element.

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

2.141.14 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.FormatY** element.

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

2.141.15 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.FormatSize** element.

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

2.141.16 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.CurrencyLanguageX** element.

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

2.141.17 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.CurrencyLanguageY** element.

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

2.141.18 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
ChartDataPointValues

The following is the XML Schema definition of the **ChartDataPointValues.CurrencyLanguageSize** element.

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

2.142 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
ChartEmptyPoints.ActionInfo

Child elements
ChartEmptyPoints.AxisLabel
ChartEmptyPoints.ChartDataLabel
ChartEmptyPoints.ChartMarker
ChartEmptyPoints.CustomProperties
ChartEmptyPoints.Style
ChartEmptyPoints.ToolTip

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>
```

2.142.1 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
ChartEmptyPoints

The following is the XML Schema definition of the **ChartEmptyPoints.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.142.2 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.

Parent elements
ChartEmptyPoints

The following is the XML Schema definition of the **ChartEmptyPoints.AxisLabel** element.

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

2.142.3 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.

Parent elements
ChartEmptyPoints

The following is the XML Schema definition of the **ChartEmptyPoints.ChartDataLabel** element.

```
<xsd:element name="ChartDataLabel" type="ChartDataLabelType" minOccurs="0" />
```

2.142.4 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.

Parent elements
ChartEmptyPoints

The following is the XML Schema definition of the **ChartEmptyPoints.ChartMarker** element.

```
<xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
```

2.142.5 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
ChartEmptyPoints

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

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

2.142.6 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
ChartEmptyPoints

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

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

2.142.7 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
ChartEmptyPoints

The following is the XML Schema definition of the **ChartEmptyPoints.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

2.143 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
ChartDataPoint
ChartSeries

Child elements
ChartItemInLegend.ActionInfo
ChartItemInLegend.Hidden
ChartItemInLegend.LegendText
ChartItemInLegend.ToolTip

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>
```

2.143.1 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
ChartItemInLegend

The following is the XML Schema definition of the **ChartItemInLegend.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.143.2 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
ChartItemInLegend

The following is the XML Schema definition of the **ChartItemInLegend.Hidden** element.

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

2.143.3 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
ChartItemInLegend

The following is the XML Schema definition of the **ChartItemInLegend.LegendText** element.

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

2.143.4 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
ChartItemInLegend

The following is the XML Schema definition of the **ChartItemInLegend.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

2.144 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
ChartDataPoint
ChartEmptyPoints
ChartLegendCustomItem
ChartSeries

Child elements
ChartMarker.Size
ChartMarker.Style
ChartMarker.Type

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>
```

2.144.1 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
ChartMarker

The following is the XML Schema definition of the **ChartMarker.Size** element.

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

2.144.2 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
ChartMarker

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

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

2.144.3 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
ChartMarker

The following is the XML Schema definition of the **ChartMarker.Type** element.

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

2.145 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.

Parent elements
ChartSeries

Child elements
ChartSmartLabel.AllowOutSidePlotArea
ChartSmartLabel.CalloutBackColor
ChartSmartLabel.CalloutLineAnchor
ChartSmartLabel.CalloutLineColor
ChartSmartLabel.CalloutLineStyle
ChartSmartLabel.CalloutLineWidth
ChartSmartLabel.CalloutStyle
ChartSmartLabel.ChartNoMoveDirections
ChartSmartLabel.Disabled
ChartSmartLabel.MarkerOverlapping
ChartSmartLabel.MaxMovingDistance
ChartSmartLabel.MinMovingDistance
ChartSmartLabel.ShowOverlapped

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>
```

2.145.1 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
ChartSmartLabel

The following is the XML Schema definition of the **ChartSmartLabel.AllowOutSidePlotArea** element.

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

2.145.2 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
ChartSmartLabel

The following is the XML Schema definition of the **ChartSmartLabel.CalloutBackColor** element.

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

2.145.3 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
ChartSmartLabel

The following is the XML Schema definition of the **ChartSmartLabel.CalloutLineAnchor** element.

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

2.145.4 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
ChartSmartLabel

The following is the XML Schema definition of the **ChartSmartLabel.CalloutLineColor** element.

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

2.145.5 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
ChartSmartLabel

The following is the XML Schema definition of the **ChartSmartLabel.CalloutLineStyle** element.

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

2.145.6 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
ChartSmartLabel

The following is the XML Schema definition of the **ChartSmartLabel.CalloutLineWidth** element.

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

2.145.7 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
ChartSmartLabel

The following is the XML Schema definition of the **ChartSmartLabel.CalloutStyle** element.

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

2.145.8 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
ChartSmartLabel

The following is the XML Schema definition of the **ChartSmartLabel.ChartNoMoveDirection** element.

```
<xsd:element name="ChartNoMoveDirections" type="ChartNoMoveDirectionsType"
```

```
minOccurs="0" />
```

2.145.9 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
ChartSmartLabel

The following is the XML Schema definition of the **ChartSmartLabel.Disabled** element.

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

2.145.10 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
ChartSmartLabel

The following is the XML Schema definition of the **ChartSmartLabel.MarkerOverlapping** element.

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

2.145.11 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
ChartSmartLabel

The following is the XML Schema definition of the **ChartSmartLabel.MaxMovingDistance** element.

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

2.145.12 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 0pt.

Following is the parent element of the **ChartSmartLabel.MinMovingDistance** element.

Parent elements
ChartSmartLabel

The following is the XML Schema definition of the **ChartSmartLabel.MinMovingDistance** element.

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

2.145.13 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
ChartSmartLabel

The following is the XML Schema definition of the **ChartSmartLabel.ShowOverlapped** element.

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

2.146 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>
```

2.146.1 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
ChartNoMoveDirections

The following is the XML Schema definition of the **ChartNoMoveDirections.Down** element.

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

2.146.2 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
ChartNoMoveDirections

The following is the XML Schema definition of the **ChartNoMoveDirections.DownLeft** element.

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

2.146.3 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
ChartNoMoveDirections

The following is the XML Schema definition of the **ChartNoMoveDirections.DownRight** element.

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

2.146.4 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
ChartNoMoveDirections

The following is the XML Schema definition of the **ChartNoMoveDirections.Left** element.

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

2.146.5 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
ChartNoMoveDirections

The following is the XML Schema definition of the **ChartNoMoveDirections.Right** element.

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

2.146.6 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
ChartNoMoveDirections

The following is the XML Schema definition of the **ChartNoMoveDirections.Up** element.

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

2.146.7 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
ChartNoMoveDirections

The following is the XML Schema definition of the **ChartNoMoveDirections.UpLeft** element.

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

2.146.8 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
ChartNoMoveDirections

The following is the XML Schema definition of the **ChartNoMoveDirections.UpRight** element.

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

2.147 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

Child elements
ChartHierarchy.ChartMembers

Applies to RDL 2012/01

The following is an additional child element of the **ChartHierarchy** element.

Child elements
ChartHierarchy.EnableDrilldown

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>
```

2.147.1 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.

Parent elements
ChartHierarchy

The following is the XML Schema definition of the **ChartHierarchy.ChartMembers** element.

```
<xsd:element name="ChartMembers" type="ChartMembersType" minOccurs="1"
             maxOccurs="1" />
```

2.147.2 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
ChartHierarchy

The following is the XML Schema definition of the **ChartHierarchy.EnableDrilldown** element.

```
<xsd:element name ="EnableDrilldown" type="xsd:boolean" />
```

2.148 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
ChartHierarchy
ChartMember

Child elements
ChartMembers.ChartMember

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>
```

2.148.1 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
ChartMembers

The following is the XML Schema definition of the **ChartMembers.ChartMember** element.

```
<xsd:element name="ChartMember" type="ChartMemberType" minOccurs="1"
maxOccurs="unbounded" />
```

2.149 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
ChartMembers

Child elements
ChartMember.ChartMembers
ChartMember.CustomProperties
ChartMember.DataElementName
ChartMember.DataElementOutput
ChartMember.Group

Child elements
ChartMember.Label
ChartMember.SortExpressions

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>
```

2.149.1 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
ChartMember

The following is the XML Schema definition of the **ChartMember.ChartMembers** element.

```
<xsd:element name="ChartMembers" type="ChartMembersType" minOccurs="0"
             maxOccurs="1" />
```

2.149.2 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
ChartMember

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

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0"
             maxOccurs="1" />
```

2.149.3 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
ChartMember

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

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

2.149.4 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
ChartMember

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>
```

2.149.5 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
ChartMember

The following is the XML Schema definition of the **ChartMember.Group** element.

```
<xsd:element name="Group" type="GroupType" minOccurs="0" maxOccurs="1" />
```

2.149.6 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
ChartMember

The following is the XML Schema definition of the **ChartMember.Label** element.

```
<xsd:element name="Label" type="StringLocIDType" minOccurs="1" maxOccurs="1" />
```

2.149.7 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
ChartMember

The following is the XML Schema definition of the **ChartMember.SortExpressions** element.

```
<xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0"
             maxOccurs="1" />
```

2.150 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
ChartLegends.ChartLegend

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>
```

2.150.1 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" />
```

2.151 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
ChartLegends

Attributes
ChartLegend.Name

Child elements
ChartLegend.AutoFitTextDisabled
ChartLegend.ChartElementPosition
ChartLegend.ChartLegendColumns
ChartLegend.ChartLegendTitle
ChartLegend.ColumnSeparator
ChartLegend.ColumnSeparatorColor
ChartLegend.ColumnSpacing
ChartLegend.DockOutsideChartArea
ChartLegend.DockToChartArea
ChartLegend.EquallySpacedItems
ChartLegend.HeaderSeparator
ChartLegend.HeaderSeparatorColor
ChartLegend.Hidden
ChartLegend.InterlacedRows
ChartLegend.InterlacedRowsColor
ChartLegend.Layout
ChartLegend.MaxAutoSize
ChartLegend.MinFontSize
ChartLegend.Position
ChartLegend.Reversed

Child elements
ChartLegend.Style
ChartLegend.TextWrapThreshold

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: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>
```

2.151.1 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
ChartLegend

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.151.2 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
ChartLegend

The following is the XML Schema definition of the **ChartLegend.AutoFitTextDisabled** element.

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

2.151.3 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
ChartLegend

The following is the XML Schema definition of the **ChartLegend.ChartElementPosition** element.

```
<xsd:element name="ChartElementPosition" type="ChartElementPositionType"
             minOccurs="0" />
```

2.151.4 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
ChartLegend

The following is the XML Schema definition of the **ChartLegend.ChartLegendColumns** element.

```
<xsd:element name="ChartLegendColumns" type="ChartLegendColumnsType"
             minOccurs="0" />
```

2.151.5 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
ChartLegend

The following is the XML Schema definition of the **ChartLegend.ChartLegendTitle** element.

```
<xsd:element name="ChartLegendTitle" type="ChartLegendTitleType" minOccurs="0" />
```

2.151.6 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" />
```

2.151.7 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" />
```

2.151.8 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
ChartLegend

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

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

2.151.9 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
ChartLegend

The following is the XML Schema definition of the **ChartLegend.DockOutsideChartArea** element.

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

2.151.10 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
ChartLegend

The following is the XML Schema definition of the **ChartLegend.DockToChartArea** element.

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

2.151.11 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
ChartLegend

The following is the XML Schema definition of the **ChartLegend.EquallySpacedItems** element.

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

2.151.12 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
ChartLegend

The following is the XML Schema definition of the **ChartLegend.HeaderSeparator** element.

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

2.151.13 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
ChartLegend

The following is the XML Schema definition of the **ChartLegend.HeaderSeparatorColor** element.

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

2.151.14 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
ChartLegend

The following is the XML Schema definition of the **ChartLegend.Hidden** element.

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

2.151.15 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
ChartLegend

The following is the XML Schema definition of the **ChartLegend.InterlacedRows** element.

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

2.151.16 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
ChartLegend

The following is the XML Schema definition of the **ChartLegend.InterlacedRowsColor** element.

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

2.151.17 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 ([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: 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
ChartLegend

The following is the XML Schema definition of the **ChartLegend.Layout** element.

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

2.151.18 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" />
```

2.151.19 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 RdlSize or an expression that evaluates to an **RdlSize**. 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" />
```

2.151.20 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" />
```

2.151.21 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" />
```

2.151.22 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" />
```

2.151.23 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" />
```

2.152 ChartLegendColumns

Applies to RDL 2008/01, RDL 2010/01, and RDL 2016/01

The **ChartLegendColumns** element is ignored.

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

Parent elements
ChartLegend

Child elements
ChartLegendColumns.ChartLegendColumn

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>
```

2.152.1 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
ChartLegendColumns

The following is the XML Schema definition of the **ChartLegendColumns.ChartLegendColumn** element.

```
<xsd:element name="ChartLegendColumn" type="ChartLegendColumnType"
             maxOccurs="unbounded" />
```

2.153 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
ChartLegendColumns

Attributes
ChartLegendColumn.Name

Child elements
ChartLegendColumn.ActionInfo
ChartLegendColumn.ColumnType
ChartLegendColumn.MaximumWidth
ChartLegendColumn.MinimumWidth
ChartLegendColumn.SeriesSymbolHeight
ChartLegendColumn.SeriesSymbolWidth
ChartLegendColumn.Style
ChartLegendColumn.ToolTip
ChartLegendColumn.Value

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>
```

2.153.1 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
ChartLegendColumn

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.153.2 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
ChartLegendColumn

The following is the XML Schema definition of the **ChartLegendColumn.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.153.3 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
ChartLegendColumn

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>
```

2.153.4 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
ChartLegendColumn

The following is the XML Schema definition of the **ChartLegendColumn.MaximumWidth** element.

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

2.153.5 ChartLegendColumn.MinimumWidth

Applies to RDL 2008/01, RDL 2010/01, and RDL 2016/01

The **ChartLegendColumn.MinimumWidth** element is ignored.

Parent elements
ChartLegendColumn

The following is the XML Schema definition of the **ChartLegendColumn.MinimumWidth** element.

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

2.153.6 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
ChartLegendColumn

The following is the XML Schema definition of the **ChartLegendColumn.SeriesSymbolHeight** element.

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

2.153.7 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
ChartLegendColumn

The following is the XML Schema definition of the **ChartLegendColumn.SeriesSymbolWidth** element.

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

2.153.8 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
ChartLegendColumn

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

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

2.153.9 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
ChartLegendColumn

The following is the XML Schema definition of the **ChartLegendColumn.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

2.153.10 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
ChartLegendColumn

The following is the XML Schema definition of the **ChartLegendColumn.Value** element.

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

2.154 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
ChartLegend

Child elements
ChartLegendTitle.Caption

Child elements
ChartLegendTitle.Style
ChartLegendTitle.TitleSeparator

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>
```

2.154.1 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.

Parent elements
ChartLegendTitle

The following is the XML Schema definition of the **ChartLegendTitle.Caption** element.

```
<xsd:element name="Caption" type="StringLocIDType" />
```

2.154.2 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.

Parent elements
ChartLegendTitle

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

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

2.154.3 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.

Parent elements
ChartLegendTitle

The following is the XML Schema definition of the **ChartLegendTitle.TitleSeparator** element.

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

2.155 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
Chart

Child elements
ChartTitles.ChartTitle

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>
```

2.155.1 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
ChartTitles

The following is the XML Schema definition of the **ChartTitles.ChartTitle** element.

```
<xsd:element name="ChartTitle" type="ChartTitleType" minOccurs="0" maxOccurs="unbounded" />
```

2.156 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
ChartTitles

Attributes
ChartTitle.Name

Child elements
ChartTitle.ActionInfo
ChartTitle.Caption
ChartTitle.ChartElementPosition
ChartTitle.DockOffset
ChartTitle.DockOutsideChartArea
ChartTitle.DockToChartArea
ChartTitle.Hidden
ChartTitle.Position
ChartTitle.Style
ChartTitle.TextOrientation
ChartTitle.ToolTip

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>
```

2.156.1 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" />
```

2.156.2 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" />
```

2.156.3 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
ChartTitle

The following is the XML Schema definition of the **ChartTitle.Caption** element.

```
<xsd:element name="Caption" type="StringLocIDType" minOccurs="1" />
```

2.156.4 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
ChartTitle

The following is the XML Schema definition of the **ChartTitle.ChartElementPosition** element.

```
<xsd:element name="ChartElementPosition" type="ChartElementPositionType"
             minOccurs="0" />
```

2.156.5 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
ChartTitle

The following is the XML Schema definition of the **ChartTitle.DockOffset** element.

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

2.156.6 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
ChartTitle

The following is the XML Schema definition of the **ChartTitle.DockOutsideChartArea** element.

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

2.156.7 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
ChartTitle

The following is the XML Schema definition of the **ChartTitle.DockToChartArea** element.

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

2.156.8 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" />
```

2.156.9 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 ([XMLSCHEMA2/2] 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: 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" />
```

2.156.10 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" />
```

2.156.11 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" />
```

2.156.12 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" />
```

2.157 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
ChartLegendColumnHeader.Style
ChartLegendColumnHeader.Value

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>
```

2.157.1 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.

Parent elements
ChartLegendColumnHeader

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

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

2.157.2 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.

Parent elements
ChartLegendColumnHeader

The following is the XML Schema definition of the **ChartLegendColumnHeader.Value** element.

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

2.158 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.

Child elements
ChartLegendCustomItems.ChartLegendCustomItem

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"
                 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"
```

```
                maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

2.158.1 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
ChartLegendCustomItems

The following is the XML Schema definition of the **ChartLegendCustomItems.ChartLegendCustomItem** element.

```
<xsd:element name="ChartLegendCustomItem" type="ChartLegendCustomItemType"
             maxOccurs="unbounded" />
```

2.159 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
ChartLegendCustomItem.Name

Child elements
ChartLegendCustomItem.ActionInfo
ChartLegendCustomItem.ChartLegendCustomItemCells
ChartLegendCustomItem.ChartMarker
ChartLegendCustomItem.Separator
ChartLegendCustomItem.SeparatorColor
ChartLegendCustomItem.Style
ChartLegendCustomItem.ToolTip

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>
```

2.159.1 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" />
```

2.159.2 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
ChartLegendCustomItem

The following is the XML Schema definition of the **ChartLegendCustomItem.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.159.3 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
ChartLegendCustomItemCells

The following is the XML Schema definition of the **ChartLegendCustomItem.ChartLegendCustomItemCells** element.

```
<xsd:element name="ChartLegendCustomItemCells"
             type="ChartLegendCustomItemCellsType" />
```

2.159.4 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
ChartLegendCustomItem

The following is the XML Schema definition of the **ChartLegendCustomItem.ChartMarker** element.

```
<xsd:element name="ChartMarker" type="ChartMarkerType" minOccurs="0" />
```

2.159.5 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
ChartLegendCustomItem

The following is the XML Schema definition of the **ChartLegendCustomItem.Separator** element.

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

2.159.6 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
ChartLegendCustomItem

The following is the XML Schema definition of the **ChartLegendCustomItem.SeparatorColor** element.

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

2.159.7 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
ChartLegendCustomItem

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

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

2.159.8 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.

Parent elements
ChartLegendCustomItem

The following is the XML Schema definition of the **ChartLegendCustomItem.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

2.160 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.

Parent elements
ChartLegendCustomItem

Child elements
ChartLegendCustomItemCells.ChartLegendCustomItemCell

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>
```

2.160.1 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
ChartLegendCustomItemCells

The following is the XML Schema definition of the **ChartLegendCustomItemCells.ChartLegendCustomItemCell** element.

```
<xsd:element name="ChartLegendCustomItemCell" type="ChartLegendCustomItemCellType"
             maxOccurs="unbounded" />
```

2.161 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
ChartLegendCustomItemCells

Attributes
ChartLegendCustomItemCell.Name

Child elements
ChartLegendCustomItemCell.ActionInfo
ChartLegendCustomItemCell.Alignment
ChartLegendCustomItemCell.BottomMargin
ChartLegendCustomItemCell.CellSpan
ChartLegendCustomItemCell.CellType
ChartLegendCustomItemCell.ImageHeight
ChartLegendCustomItemCell.ImageWidth
ChartLegendCustomItemCell.LeftMargin
ChartLegendCustomItemCell.RightMargin
ChartLegendCustomItemCell.Style
ChartLegendCustomItemCell.SymbolHeight
ChartLegendCustomItemCell.SymbolWidth

Child elements
ChartLegendCustomItemCell.Text
ChartLegendCustomItemCell.ToolTip
ChartLegendCustomItemCell.TopMargin

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: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>
```

2.161.1 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
ChartLegendCustomItemCell

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.161.2 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
ChartLegendCustomItemCell

The following is the XML Schema definition of the **ChartLegendCustomItemCell.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.161.3 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.

Parent elements
ChartLegendCustomItemCell

The following is the XML Schema definition of the **ChartLegendCustomItemCell.Alignment** element.

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

2.161.4 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.

Parent elements
ChartLegendCustomItemCell

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

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

2.161.5 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.

Parent elements
ChartLegendCustomItemCell

The following is the XML Schema definition of the **ChartLegendCustomItemCell.CellSpan** element.

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

2.161.6 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.

Parent elements
ChartLegendCustomItemCell

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>
```

2.161.7 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
ChartLegendCustomItemCell

The following is the XML Schema definition of the **ChartLegendCustomItemCell.ImageHeight** element.

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

2.161.8 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
ChartLegendCustomItemCell

The following is the XML Schema definition of the **ChartLegendCustomItemCell.ImageWidth** element.

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

2.161.9 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.

Parent elements
ChartLegendCustomItemCell

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

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

2.161.10 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.

Parent elements
ChartLegendCustomItemCell

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

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

2.161.11 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.

Parent elements
ChartLegendCustomItemCell

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

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

2.161.12 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.

Parent elements
ChartLegendCustomItemCell

The following is the XML Schema definition of the **ChartLegendCustomItemCell.SymbolHeight** element.

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

2.161.13 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
ChartLegendCustomItemCell

The following is the XML Schema definition of the **ChartLegendCustomItemCell.SymbolWidth** element.

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

2.161.14 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
ChartLegendCustomItemCell

The following is the XML Schema definition of the **ChartLegendCustomItemCell.Text** element.

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

2.161.15 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
ChartLegendCustomItemCell

The following is the XML Schema definition of the **ChartLegendCustomItemCell.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

2.161.16 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
ChartLegendCustomItemCell

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

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

2.162 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
ReportItems
CellContents
CustomReportItem.AltReportItem

Attributes
GaugePanel.Name

Child elements
GaugePanel.Style
GaugePanel.ActionInfo
GaugePanel.Bookmark
GaugePanel.CustomProperties
GaugePanel.DataElementName
GaugePanel.DataElementOutput
GaugePanel.DocumentMapLabel
GaugePanel.Height

Child elements
GaugePanel.Left
GaugePanel.RepeatWith
GaugePanel.ToolTip
GaugePanel.Top
GaugePanel.Visibility
GaugePanel.Width
GaugePanel.ZIndex
GaugePanel.DataSetName
GaugePanel.Filters
GaugePanel.NoRowsMessage
GaugePanel.PageBreak
GaugePanel.PageName
GaugePanel.SortExpressions
GaugePanel.AntiAliasing
GaugePanel.AutoLayout
GaugePanel.BackFrame
GaugePanel.GaugeImages
GaugePanel.GaugeLabels
GaugePanel.GaugeMember
GaugePanel.LinearGauges
GaugePanel.NumericIndicators
GaugePanel.RadialGauges
GaugePanel.ShadowIntensity
GaugePanel.StateIndicators
GaugePanel.TextAntiAliasingQuality
GaugePanel.TopImage

Applies to RDL 2011/01

Child elements
GaugePanel.Relationship

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>
```

2.162.1 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" />
```

2.162.2 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">
```

2.162.3 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
GaugePanel

The following is the XML Schema definition of the **GaugePanel.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0">
```

2.162.4 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">
```

2.162.5 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" />
```

2.162.6 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">
```

2.162.7 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>
```

2.162.8 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" />
```

2.162.9 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
GaugePanel

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

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

2.162.10 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
GaugePanel

The following is the XML Schema definition of the **GaugePanel.Left** element.

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

2.162.11 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
GaugePanel

The following is the XML Schema definition of the **GaugePanel.RepeatWith** element.

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

2.162.12 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">
```

2.162.13 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">
```

2.162.14 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">
```

2.162.15 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">
```

2.162.16 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">
```

2.162.17 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">
```

2.162.18 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" />
```

2.162.19 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">
```

2.162.20 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">
```

2.162.21 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
GaugePanel

The following is the XML Schema definition of the **GaugePanel.PageName** element.

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

2.162.22 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" />
```

2.162.23 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">
```

2.162.24 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">
```

2.162.25 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">
```

2.162.26 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
GaugePanel

The following is the XML Schema definition of the **GaugePanel.GaugeImages** element.

```
<xsd:element name="GaugeImages" type="GaugeImagesType" minOccurs="0" />
```

2.162.27 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">
```

2.162.28 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">
```

2.162.29 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
GaugePanel

The following is the XML Schema definition of the **GaugePanel.LinearGauges** element.

```
<xsd:element name="LinearGauges" type="LinearGaugesType" minOccurs="0">
```

2.162.30 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
GaugePanel

The following is the XML Schema definition of the **GaugePanel.NumericIndicators** element.

```
<xsd:element name="NumericIndicators" type="NumericIndicatorsType"
             minOccurs="0" />
```

2.162.31 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
GaugePanel

The following is the XML Schema definition of the **GaugePanel.RadialGauges** element.

```
<xsd:element name="RadialGauges" type="RadialGaugesType" minOccurs="0">
```

2.162.32 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
GaugePanel

The following is the XML Schema definition of the **GaugePanel.ShadowIntensity** element.

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

2.162.33 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
GaugePanel

The following is the XML Schema definition of the **GaugePanel.StateIndicators** element.

```
<xsd:element name="StateIndicators" type="StateIndicatorsType" minOccurs="0" />
```

2.162.34 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
GaugePanel

The following is the XML Schema definition of the **GaugePanel.TextAntiAliasingQuality** element.

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

2.162.35 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
GaugePanel

The following is the XML Schema definition of the **GaugePanel.TopImage** element.

```
<xsd:element name="TopImage" type="TopImageType" minOccurs="0">
```

2.162.36 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
GaugePanel

The following is the XML Schema definition of the **GaugePanel.Relationship** element.

```
<xsd:element name ="Relationship" type="RelationshipType" minOccurs="0" />
```

2.163 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
LinearGauge
RadialGauge
GaugePanel

Child elements
BackFrame.FrameBackground
BackFrame.FrameImage
BackFrame.FrameShape
BackFrame.FrameStyle
BackFrame.FrameWidth
BackFrame.GlassEffect
BackFrame.Style

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: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>
```

2.163.1 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" />
```

2.163.2 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" />
```

2.163.3 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" />
```

2.163.4 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" />
```

2.163.5 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" />
```

2.163.6 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" />
```

2.163.7 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" />
```

2.164 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
BackFrame

Child elements
FrameBackground.Style

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>
```

2.164.1 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
FrameBackground

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

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

2.165 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
BackFrame

Child elements
FrameImage.MIMEType
FrameImage.Source
FrameImage.TransparentColor
FrameImage.Value
FrameImage.ClipImage
FrameImage.HueColor
FrameImage.Transparency

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: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>
```

2.165.1 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" />
```

2.165.2 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" "/>
```

2.165.3 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" />
```

2.165.4 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
FrameImage

The following is the XML Schema definition of the **FrameImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1" "/>
```

2.165.5 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
FrameImage

The following is the XML Schema definition of the **FrameImage.ClipImage** element.

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

2.165.6 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
FrameImage

The following is the XML Schema definition of the **FrameImage.HueColor** element.

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

2.165.7 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
FrameImage

The following is the XML Schema definition of the **FrameImage.Transparency** element.

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

2.166 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
GaugePanel

Child elements
GaugeImages.GaugeImage

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>
```

2.166.1 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
GaugeImage

The following is the XML Schema definition of the **GaugeImages.GaugeImage** element.

```
<xsd:element name="GaugeImage" type="GaugeImageType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.167 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
GaugeImages

Attributes
GaugeImage.Name

Child elements
GaugeImage.ActionInfo
GaugeImage.Height
GaugeImage.Hidden
GaugeImage.Left
GaugeImage.ParentItem
GaugeImage.ToolTip
GaugeImage.Top

Child elements
GaugeImage.Width
GaugeImage.ZIndex
GaugeImage.Angle
GaugeImage.MIMEType
GaugeImage.ResizeMode
GaugeImage.Source
GaugeImage.Transparency
GaugeImage.TransparentColor
GaugeImage.Value

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: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>
```

2.167.1 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
GaugeImage

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.167.2 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
GaugeImage

The following is the XML Schema definition of the **GaugeImage.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.167.3 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.

Parent elements
GaugeImage

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

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

2.167.4 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.

Parent elements
GaugeImage

The following is the XML Schema definition of the **GaugeImage.Hidden** element.

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

2.167.5 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.

Parent elements
GaugeImage

The following is the XML Schema definition of the **GaugeImage.Left** element.

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

2.167.6 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.

Parent elements
GaugeImage

The following is the XML Schema definition of the **GaugeImage.ParentItem** element.

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

2.167.7 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.

Parent elements
GaugeImage

The following is the XML Schema definition of the **GaugeImage.ToolTip** element.

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

2.167.8 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.

Parent elements
GaugeImage

The following is the XML Schema definition of the **GaugeImage.Top** element.

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

2.167.9 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.

Parent elements
GaugeImage

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

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

2.167.10 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.

Parent elements
GaugeImage

The following is the XML Schema definition of the **GaugeImage.ZIndex** element.

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

2.167.11 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.

Parent elements
GaugeImage

The following is the XML Schema definition of the **GaugeImage.Angle** element.

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

2.167.12 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.

Parent elements
GaugeImage

The following is the XML Schema definition of the **GaugeImage.MIMEType** element.

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

2.167.13 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.

Parent elements
GaugeImage

The following is the XML Schema definition of the **GaugeImage.ResizeMode** element.

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

2.167.14 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.

Parent elements
GaugeImage

The following is the XML Schema definition of the **GaugeImage.Source** element.

```
<xsd:element name="Source" type="xsd:string" minOccurs="1" />
```

2.167.15 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
GaugeImage

The following is the XML Schema definition of the **GaugeImage.Transparency** element.

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

2.167.16 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
GaugeImage

The following is the XML Schema definition of the **GaugeImage.TransparentColor** element.

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

2.167.17 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
GaugeImage

The following is the XML Schema definition of the **GaugeImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1" />
```

2.168 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>
```

2.168.1 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" />
```

2.169 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
GaugeLabels

Attributes
GaugeLabel.Name

Child elements
GaugeLabel.ActionInfo
GaugeLabel.Height
GaugeLabel.Hidden
GaugeLabel.Left
GaugeLabel.ParentItem
GaugeLabel.ToolTip
GaugeLabel.Top
GaugeLabel.Width
GaugeLabel.ZIndex
GaugeLabel.Angle
GaugeLabel.ResizeMode
GaugeLabel.Style
GaugeLabel.Text
GaugeLabel.TextShadowOffset
GaugeLabel.UseFontPercent

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" typ="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" typ="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>
```

2.169.1 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" />
```

2.169.2 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" />
```

2.169.3 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" />
```

2.169.4 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
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.Hidden** element.

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

2.169.5 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
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.Left** element.

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

2.169.6 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
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.ParentItem** element.

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

2.169.7 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
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.ToolTip** element.

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

2.169.8 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
GaugeLabel

The following is the XML Schema definition of the **GaugeLabel.Top** element.

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

2.169.9 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" />
```

2.169.10 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" />
```

2.169.11 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">
```

2.169.12 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">
```

2.169.13 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">
```

2.169.14 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">
```

2.169.15 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">
```

2.169.16 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">
```

2.170 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.

Parent elements
GaugePanel
GaugeMember

Child elements
GaugeMember.GaugeMember
GaugeMember.Group
GaugeMember.SortExpressions

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>
```

2.170.1 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.

Parent elements
GaugeMember

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

```
<xsd:element name="GaugeMember" type="GaugeMemberType" minOccurs="0">
```

2.170.2 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
GaugeMember

The following is the XML Schema definition of the **GaugeMember.Group** element.

```
<xsd:element name="Group" type="GroupType" minOccurs="1">
```

2.170.3 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
GaugeMember

The following is the XML Schema definition of the **GaugeMember.SortExpressions** element.

```
<xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
```

2.171 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>
```

2.171.1 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" />
```

2.172 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
LinearGauge.Name

Child elements
LinearGauge.ActionInfo
LinearGauge.Height
LinearGauge.Hidden
LinearGauge.Left
LinearGauge.ParentItem
LinearGauge.ToolTip
LinearGauge.Top
LinearGauge.Width
LinearGauge.ZIndex
LinearGauge.AspectRatio
LinearGauge.BackFrame
LinearGauge.ClipContent
LinearGauge.GaugeScales
LinearGauge.TopImage
LinearGauge.Orientation

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>
```

2.172.1 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" />
```

2.172.2 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" />
```

2.172.3 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" />
```

2.172.4 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" />
```

2.172.5 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
LinearGauge

The following is the XML Schema definition of the **LinearGauge.Left** element.

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

2.172.6 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
LinearGauge

The following is the XML Schema definition of the **LinearGauge.ParentItem** element.

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

2.172.7 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" />
```

2.172.8 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
LinearGauge

The following is the XML Schema definition of the **LinearGauge.Top** element.

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

2.172.9 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" />
```

2.172.10 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" />
```

2.172.11 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" />
```

2.172.12 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" />
```

2.172.13 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" />
```

2.172.14 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">
```

2.172.15 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" />
```

2.172.16 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" />
```

2.173 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

Child elements
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>
```

2.173.1 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.

Parent elements
LinearScales

The following is the XML Schema definition of the **LinearScales.LinearScale** element.

```
<xsd:element name="LinearScale" type="LinearScaleType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.174 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.

Parent elements
LinearScales

Attributes
LinearScale.Name

Child elements
LinearScale.ActionInfo
LinearScale.CustomLabels
LinearScale.GaugeMajorTickMarks
LinearScale.GaugeMinorTickMarks
LinearScale.GaugePointers
LinearScale.Hidden
LinearScale.Interval
LinearScale.IntervalOffset
LinearScale.Logarithmic
LinearScale.LogarithmicBase
LinearScale.MaximumPin
LinearScale.MaximumValue
LinearScale.MinimumPin
LinearScale.MinimumValue
LinearScale.Multiplier
LinearScale.Reversed
LinearScale.ScaleLabels
LinearScale.ScaleRanges
LinearScale.Style
LinearScale.TickMarksOnTop
LinearScale.ToolTip
LinearScale.Width
LinearScale.EndMargin
LinearScale.Position
LinearScale.StartMargin

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>
```

2.174.1 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" />
```

2.174.2 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" />
```

2.174.3 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.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.CustomLabels** element.

```
<xsd:element name="CustomLabels" type="CustomLabelsType" minOccurs="0" />
```

2.174.4 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.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.GaugeMajorTickMarks** element.

```
<xsd:element name="GaugeMajorTickMarks" type="GaugeTickMarksType" minOccurs="0" />
```

2.174.5 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.

Parent elements
LinearScale

The following is the XML Schema definition of the **LinearScale.GaugeMinorTickMarks** element.

```
<xsd:element name="GaugeMinorTickMarks"
             type="GaugeTickMarksType" minOccurs="0" />
```

2.174.6 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" />
```

2.174.7 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" />
```

2.174.8 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^{+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" />
```

2.174.9 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" />
```

2.174.10 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" />
```

2.174.11 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 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" />
```

2.174.12 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" />
```

2.174.13 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" />
```

2.174.14 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 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" />
```

2.174.15 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" />
```

2.174.16 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" />
```

2.174.17 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" />
```

2.174.18 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" />
```

2.174.19 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" />
```

2.174.20 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" />
```

2.174.21 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" />
```

2.174.22 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" />
```

2.174.23 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" />
```

2.174.24 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">
```

2.174.25 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"
```

2.174.26 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">
```

2.175 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>
```

2.175.1 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" />
```

2.176 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
CustomLabel.Name

Child elements
CustomLabel.AllowUpsideDown
CustomLabel.DistanceFromScale
CustomLabel.FontAngle
CustomLabel.Hidden
CustomLabel.Placement
CustomLabel.RotateLabel
CustomLabel.Style
CustomLabel.Text
CustomLabel.TickMarkStyle
CustomLabel.UseFontPercent
CustomLabel.Value

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:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
```

2.176.1 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" />
```

2.176.2 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" />
```

2.176.3 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" />
```

2.176.4 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" />
```

2.176.5 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" />
```

2.176.6 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" />
```

2.176.7 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" />
```

2.176.8 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" />
```

2.176.9 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" />
```

2.176.10 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" />
```

2.176.11 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
CustomLabel

The following is the XML Schema definition of the **CustomLabel.UseFontPercent** element.

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

2.176.12 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
CustomLabel

The following is the XML Schema definition of the **CustomLabel.Value** element.

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

2.177 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 for the **TickMarkStyle** element.

Parent elements
CustomLabel

Child elements
TickMarkStyle.DistanceFromScale

Child elements
TickMarkStyle.EnableGradient
TickMarkStyle.GradientDensity
TickMarkStyle.Hidden
TickMarkStyle.Length
TickMarkStyle.Placement
TickMarkStyle.Shape
TickMarkStyle.Style
TickMarkStyle.TickMarkImage
TickMarkStyle.Width

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>
```

2.177.1 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
TickMarkStyle

The following is the XML Schema definition of the **TickMarkStyle.DistanceFromScale** element.

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

2.177.2 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
TickMarkStyle

The following is the XML Schema definition of the **TickMarkStyle.EnableGradient** element.

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

2.177.3 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
TickMarkStyle

The following is the XML Schema definition of the **TickMarkStyle.GradientDensity** element.

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

2.177.4 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
TickMarkStyle

The following is the XML Schema definition of the **TickMarkStyle.Hidden** element.

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

2.177.5 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
TickMarkStyle

The following is the XML Schema definition of the **TickMarkStyle.Length** element.

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

2.177.6 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
TickMarkStyle

The following is the XML Schema definition of the **TickMarkStyle.Placement** element.

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

2.177.7 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
TickMarkStyle

The following is the XML Schema definition of the **TickMarkStyle.Shape** element.

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

2.177.8 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
TickMarkStyle

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

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

2.177.9 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
TickMarkStyle

The following is the XML Schema definition of the **TickMarkStyle.TickMarkImage** element.

```
<xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0">
```

2.177.10 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
TickMarkStyle

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

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

2.178 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
LinearScale
LinearPointer
ScaleRange
NumericIndicator
NumericIndicatorRange
RadialScale
RadialPointer
StateIndicator
IndicatorState

Child elements
GaugeInputValue.AddConstant
GaugeInputValue.DataElementName
GaugeInputValue.DataElementOutput
GaugeInputValue.Formula
GaugeInputValue.MaxPercent
GaugeInputValue.MinPercent
GaugeInputValue.Multiplier
GaugeInputValue.Value

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: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>
```

2.178.1 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.

Parent elements
GaugeInputValue

The following is the XML Schema definition of the **GaugeInputValue.AddConstant** element.

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

2.178.2 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">
```

2.178.3 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>
```

2.178.4 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
GaugeInputValue

The following is the XML Schema definition of the **GaugeInputValue.Formula** element.

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

2.178.5 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.

Parent elements
GaugeInputValue

The following is the XML Schema definition of the **GaugeInputValue.MaxPercent** element.

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

2.178.6 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.

Parent elements
GaugeInputValue

The following is the XML Schema definition of the **GaugeInputValue.MinPercent** element.

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

2.178.7 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.

Parent elements
GaugeInputValue

The following is the XML Schema definition of the **GaugeInputValue.Multiplier** element.

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

2.178.8 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
GaugeInputValue

The following is the XML Schema definition of the **GaugeInputValue.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1">
```

2.179 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
LinearScale
RadialScale

Child elements
GaugeTickMarks.Interval
GaugeTickMarks.IntervalOffset
GaugeTickMarks.DistanceFromScale
GaugeTickMarks.EnableGradient
GaugeTickMarks.GradientDensity
GaugeTickMarks.Hidden
GaugeTickMarks.Length

Child elements
GaugeTickMarks.Placement
GaugeTickMarks.Shape
GaugeTickMarks.Style
GaugeTickMarks.TickMarkImage
GaugeTickMarks.Width

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>
```

2.179.1 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
GaugeTickMarks

The following is the XML Schema definition of the **GaugeTickMarks.Interval** element.

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

2.179.2 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
GaugeTickMarks

The following is the XML Schema definition of the **GaugeTickMarks.IntervalOffset** element.

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

2.179.3 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
GaugeTickMarks

The following is the XML Schema definition of the **GaugeTickMarks.DistanceFromScale** element.

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

2.179.4 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
GaugeTickMarks

The following is the XML Schema definition of the **GaugeTickMarks.EnableGradient** element.

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

2.179.5 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
GaugeTickMarks

The following is the XML Schema definition of the **GaugeTickMarks.GradientDensity** element.

```
xsd:element name="GradientDensity" type="xsd:string" minOccurs="0">
```

2.179.6 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
GaugeTickMarks

The following is the XML Schema definition of the **GaugeTickMarks.Hidden** element.

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

2.179.7 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
GaugeTickMarks

The following is the XML Schema definition of the **GaugeTickMarks.Length** element.

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

2.179.8 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
GaugeTickMarks

The following is the XML Schema definition of the **GaugeTickMarks.Placement** element.

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

2.179.9 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
GaugeTickMarks

The following is the XML Schema definition of the **GaugeTickMarks.Shape** element.

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

2.179.10 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
GaugeTickMarks

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

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

2.179.11 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
GaugeTickMarks

The following is the XML Schema definition of the **GaugeTickMarks.TickMarkImage** element.

```
<xsd:element name="TickMarkImage" type="TopImageType" minOccurs="0">
```

2.179.12 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
GaugeTickMarks

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

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

2.180 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>
```

2.180.1 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" />
```

2.181 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
LinearPointers

Attributes
LinearPointer.Name

Child elements
LinearPointer.Thermometer
LinearPointer.Type
LinearPointer.ActionInfo
LinearPointer.BarStart
LinearPointer.DistanceFromScale
LinearPointer.GaugeInputValue
LinearPointer.Hidden
LinearPointer.MarkerLength
LinearPointer.MarkerStyle
LinearPointer.Placement
LinearPointer.PointerImage
LinearPointer.SnappingEnabled
LinearPointer.SnappingInterval
LinearPointer.Style
LinearPointer.ToolTip
LinearPointer.Width

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: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>
```

2.181.1 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.

Parent elements
LinearPointer

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.181.2 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.

Parent elements
LinearPointer

The following is the XML Schema definition of the **LinearPointer.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.181.3 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 ([XMLSCHEMA2/2] 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.

Parent elements
LinearPointer

The following is the XML Schema definition of the **LinearPointer.BarStart** element.

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

2.181.4 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" />
```

2.181.5 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" />
```

2.181.6 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" />
```

2.181.7 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" />
```

2.181.8 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" />
```

2.181.9 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" />
```

2.181.10 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" />
```

2.181.11 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
LinearPointer

The following is the XML Schema definition of the **LinearPointer.SnappingEnabled** element.

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

2.181.12 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
LinearPointer

The following is the XML Schema definition of the **LinearPointer.SnappingInterval** element.

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

2.181.13 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
LinearPointer

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

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

2.181.14 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
LinearPointer

The following is the XML Schema definition of the **LinearPointer.ToolTip** element.

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

2.181.15 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
LinearPointer

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

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

2.181.16 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
LinearPointer

The following is the XML Schema definition of the **LinearPointer.Thermometer** element.

```
<xsd:element name="Thermometer" type="ThermometerType" minOccurs="0">
```

2.181.17 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
LinearPointer

The following is the XML Schema definition of the **LinearPointer.Type** element.

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

2.182 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
LinearPointer
RadialPointer

Child elements
PointerImage.HueColor
PointerImage.OffsetX
PointerImage.OffsetY
PointerImage.Transparency
PointerImage.MIMEType
PointerImage.Source
PointerImage.TransparentColor

Child elements
PointerImage.Value

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>
```

2.182.1 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.

Parent elements
PointerImage

The following is the XML Schema definition of the **PointerImage.MIMEType** element.

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

2.182.2 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">
```

2.182.3 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">
```

2.182.4 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">
```

2.182.5 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">
```

2.182.6 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">
```

2.182.7 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">
```

2.182.8 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">
```

2.183 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 for the **Thermometer** element.

Parent elements
LinearPointer

Child elements
Thermometer.BulbOffset
Thermometer.BulbSize
Thermometer.Style
Thermometer.ThermometerStyle

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>
```

2.183.1 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.

Parent elements
Thermometer

The following is the XML Schema definition of the **Thermometer.BulbOffset** element.

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

2.183.2 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.

Parent elements
Thermometer

The following is the XML Schema definition of the **Thermometer.BulbSize** element.

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

2.183.3 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.

Parent elements
Thermometer

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

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

2.183.4 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
Thermometer

The following is the XML Schema definition of the **Thermometer.ThermometerStyle** element.

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

2.184 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
LinearScale
RadialScale

Child elements
ScaleLabels.AllowUpsideDown
ScaleLabels.DistanceFromScale
ScaleLabels.FontAngle
ScaleLabels.Hidden
ScaleLabels.Interval
ScaleLabels.IntervalOffset
ScaleLabels.Placement
ScaleLabels.RotateLabels
ScaleLabels.ShowEndLabels
ScaleLabels.Style

Child elements
ScaleLabels.UseFontPercent

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>
```

2.184.1 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.

Parent elements
ScaleLabels

The following is the XML Schema definition of the **ScaleLabels.AllowUpsideDown** element.

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

2.184.2 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
ScaleLabels

The following is the XML Schema definition of the **ScaleLabels.DistanceFromScale** element.

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

2.184.3 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
ScaleLabels

The following is the XML Schema definition of the **ScaleLabels.FontAngle** element.

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

2.184.4 ScaleLabels.Hidden

Applies to RDL 2008/01, RDL 2010/01, and RDL 2016/01

The **ScaleLabels.Hidden** element specifies whether the labels within a RadialScaleor 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
ScaleLabels

The following is the XML Schema definition of the **ScaleLabels.Hidden** element.

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

2.184.5 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
ScaleLabels

The following is the XML Schema definition of the **ScaleLabels.Interval** element.

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

2.184.6 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
ScaleLabels

The following is the XML Schema definition of the **ScaleLabels.IntervalOffset** element.

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

2.184.7 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
ScaleLabels

The following is the XML Schema definition of the **ScaleLabels.Placement** element.

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

2.184.8 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">
```

2.184.9 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
ScaleLabels

The following is the XML Schema definition of the **ScaleLabels.ShowEndLabels** element.

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

2.184.10 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
ScaleLabels

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

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

2.184.11 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
ScaleLabels

The following is the XML Schema definition of the **ScaleLabels.UseFontPercent** element.

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

2.185 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
LinearScale
RadialScale

Child elements
ScalePin.DistanceFromScale
ScalePin.EnableGradient
ScalePin.GradientDensity
ScalePin.Hidden
ScalePin.Length
ScalePin.Placement
ScalePin.Shape
ScalePin.Style
ScalePin.TickMarkImage
ScalePin.Width
ScalePin.Enable
ScalePin.Location
ScalePin.PinLabel

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" />
```

```
      <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>
```

2.185.1 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">
```

2.185.2 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" />
```

2.185.3 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">
```

2.185.4 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">
```

2.185.5 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">
```

2.185.6 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">
```

2.185.7 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">
```

2.185.8 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">
```

2.185.9 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">
```

2.185.10 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">
```

2.185.11 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">
```

2.185.12 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">
```

2.185.13 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">
```

2.186 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
PinLabel.AllowUpsideDown
PinLabel.DistanceFromScale
PinLabel.FontAngle
PinLabel.Placement
PinLabel.RotateLabel
PinLabel.Style
PinLabel.Text
PinLabel.UseFontPercent

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>
```

2.186.1 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 ([XMLSCHEMA2/2] 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.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">
```

2.186.2 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">
```

2.186.3 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">
```

2.186.4 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">
```

2.186.5 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">
```

2.186.6 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">
```

2.186.7 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">
```

2.186.8 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">
```

2.187 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 for 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>
```

2.187.1 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
ScaleRanges

The following is the XML Schema definition of the **ScaleRanges.ScaleRange** element.

```
<xsd:element name="ScaleRange" type="ScaleRangeType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.188 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
ScaleRanges

Attributes
ScaleRange.Name

Child elements
ScaleRange.ActionInfo
ScaleRange.BackgroundGradientType
ScaleRange.DistanceFromScale
ScaleRange.EndValue
ScaleRange.EndWidth
ScaleRange.Hidden
ScaleRange.InRangeBarPointerColor
ScaleRange.InRangeLabelColor
ScaleRange.InRangeTickMarksColor

Child elements
ScaleRange.Placement
ScaleRange.StartValue
ScaleRange.StartWidth
ScaleRange.Style
ScaleRange.ToolTip

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>
```

2.188.1 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" />
```

2.188.2 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">
```

2.188.3 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" />
```

2.188.4 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">
```

2.188.5 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">
```

2.188.6 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">
```

2.188.7 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">
```

2.188.8 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" />
```

2.188.9 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">
```

2.188.10 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" />
```

2.188.11 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 ([XMLSCHEMA2/2] 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 **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">
```

2.188.12 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">
```

2.188.13 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">
```

2.188.14 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">
```

2.188.15 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">
```

2.189 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 for the **TopImage** element.

Parent elements
GaugePanel
LinearGauge
TickMarkStyle
GaugeTickMarks
ScalePin
RadialGauge

Child elements
TopImage.MIMEType
TopImage.Source
TopImage.TransparentColor
TopImage.Value
TopImage.HueColor

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>
```

2.189.1 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">
```

2.189.2 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">
```

2.189.3 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">
```

2.189.4 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
TopImage

The following is the XML Schema definition of the **TopImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1">
```

2.189.5 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">
```

2.190 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
GaugePanel

Child elements
NumericIndicator

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>
```

2.190.1 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
NumericIndicators

The following is the XML Schema definition of the **NumericIndicators.NumericIndicator** element.

```
<xsd:element name="NumericIndicator" type="NumericIndicatorType" minOccurs="1"
```

```
                maxOccurs="unbounded" />
```

2.191 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
NumericIndicators

Attributes
NumericIndicator.Name

Child elements
NumericIndicator.ActionInfo
NumericIndicator.Height
NumericIndicator.Hidden
NumericIndicator.Left
NumericIndicator.ParentItem
NumericIndicator.ToolTip
NumericIndicator.Top
NumericIndicator.Width
NumericIndicator.ZIndex
NumericIndicator.DecimalDigitColor
NumericIndicator.DecimalDigits
NumericIndicator.DigitColor
NumericIndicator.Digits
NumericIndicator.GaugeInputValue
NumericIndicator.IndicatorStyle
NumericIndicator.LedDimColor
NumericIndicator.MaximumValue
NumericIndicator.MinimumValue

Child elements
NumericIndicator.Multiplier
NumericIndicator.NumericIndicatorRanges
NumericIndicator.Offstring
NumericIndicator.OutOfRangeString
NumericIndicator.ResizeMode
NumericIndicator.SeparatorColor
NumericIndicator.SeparatorWidth
NumericIndicator.ShowDecimalPoint
NumericIndicator.ShowLeadingZeros
NumericIndicator.ShowSign
NumericIndicator.SnappingEnabled
NumericIndicator.SnappingInterval
NumericIndicator.Style
NumericIndicator.UseFontPercent

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>
```

2.191.1 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
NumericIndicator

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.191.2 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.191.3 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
NumericIndicator

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

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

2.191.4 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.Hidden** element.

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

2.191.5 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.Left** element.

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

2.191.6 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.ParentItem** element.

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

2.191.7 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.ToolTip** element.

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

2.191.8 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.Top** element.

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

2.191.9 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
NumericIndicator

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

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

2.191.10 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.ZIndex** element.

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

2.191.11 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.DecimalDigitColor** element.

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

2.191.12 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.DecimalDigits** element.

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

2.191.13 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.DigitColor** element.

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

2.191.14 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.Digits** element.

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

2.191.15 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.GaugeInputValue** element.

```
<xsd:element name="GaugeInputValue" type="GaugeInputValueType" minOccurs="1" />
```

2.191.16 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.IndicatorStyle** element.

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

2.191.17 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.LedDimColor** element.

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

2.191.18 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.MaximumValue** element.

```
<xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
```

2.191.19 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.MinimumValue** element.

```
<xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0" />
```

2.191.20 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.Multiplier** element.

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

2.191.21 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.NumericIndicatorRanges** element.

```
<xsd:element name="NumericIndicatorRanges" type="NumericIndicatorRangesType"
             minOccurs="0" />
```

2.191.22 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.Offstring** element.

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

2.191.23 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.OutOfRangeString** element.

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

2.191.24 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.ResizeMode** element.

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

2.191.25 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.SeparatorColor** element.

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

2.191.26 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.SeparatorWidth** element.

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

2.191.27 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.ShowDecimalPoint** element.

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

2.191.28 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.ShowLeadingZeros** element.

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

2.191.29 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.ShowSign** element.

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

2.191.30 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.SnappingEnabled** element.

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

2.191.31 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.SnappingInterval** element.

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

2.191.32 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
NumericIndicator

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

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

2.191.33 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
NumericIndicator

The following is the XML Schema definition of the **NumericIndicator.UseFontPercent** element.

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

2.192 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
NumericIndicator

Child elements
NumericIndicatorRanges.NumericIndicatorRange

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>
```

2.192.1 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 **NumericIndicatorsRanges.NumericIndicatorRange** element.

Parent elements
NumericIndicatorRanges

The following is the XML Schema definition of the **NumericIndicatorRanges.NumericIndicatorRange** element.

```
<xsd:element name="NumericIndicatorRange" type="NumericIndicatorRangeType"
```

```
            minOccurs="1" maxOccurs="unbounded" />
```

2.193 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
NumericIndicatorRanges

Attributes
NumericIndicatorRange.Name

Child elements
NumericIndicatorRange.StartValue
NumericIndicatorRange.EndValue
NumericIndicatorRange.DecimalDigitColor
NumericIndicatorRange.DigitColor

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>
```

2.193.1 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
NumericIndicatorRange

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" /
```

2.193.2 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
NumericIndicatorRange

The following is the XML Schema definition of the **NumericIndicatorRange.StartValue** element.

```
<xsd:element name="StartValue" type="GaugeInputValueType" minOccurs="0" />
```

2.193.3 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
NumericIndicatorRange

The following is the XML Schema definition of the **NumericIndicatorRange.EndValue** element.

```
<xsd:element name="EndValue" type="GaugeInputValueType" minOccurs="0" />
```

2.193.4 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
NumericIndicatorRange

The following is the XML Schema definition of the **NumericIndicatorRange.DecimalDigitColor** element.

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

2.193.5 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
NumericIndicatorRange

The following is the XML Schema definition of the **NumericIndicatorRange.DigitColor** element.

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

2.194 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

Child elements
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>
```

2.194.1 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.

Parent elements
RadialGauges

The following is the XML Schema definition of the **RadialGauges.RadialGauge** element.

```
<xsd:element name="RadialGauge" type="RadialGaugeType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.195 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.

Parent elements
RadialGauges

Attributes
RadialGauge.Name

Child elements
RadialGauge.PivotX
RadialGauge.PivotY
RadialGauge.AspectRatio
RadialGauge.BackFrame
RadialGauge.ClipContent
RadialGauge.GaugeScales
RadialGauge.TopImage
RadialGauge.ActionInfo
RadialGauge.Height
RadialGauge.Hidden
RadialGauge.Left
RadialGauge.ParentItem
RadialGauge.ToolTip
RadialGauge.Top
RadialGauge.Width
RadialGauge.ZIndex

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"
```

```
                    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>
```

2.195.1 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" />
```

2.195.2 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" />
```

2.195.3 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" />
```

2.195.4 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" />
```

2.195.5 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
RadialGauge

The following is the XML Schema definition of the **RadialGauge.Left** element.

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

2.195.6 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
RadialGauge

The following is the XML Schema definition of the **RadialGauge.ParentItem** element.

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

2.195.7 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" />
```

2.195.8 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
RadialGauge

The following is the XML Schema definition of the **RadialGauge.Top** element.

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

2.195.9 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" />
```

2.195.10 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" />
```

2.195.11 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" />
```

2.195.12 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" />
```

2.195.13 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" />
```

2.195.14 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" />
```

2.195.15 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.

Parent elements
RadialGauge

The following is the XML Schema definition of the **RadialGauge.TopImage** element.

```
<xsd:element name="TopImage" type="TopImageType" minOccurs="0" />
```

2.195.16 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.

Parent elements
RadialGauge

The following is the XML Schema definition of the **RadialGauge.PivotX** element.

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

2.195.17 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.

Parent elements
RadialGauge

The following is the XML Schema definition of the **RadialGauge.PivotY** element.

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

2.196 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>
```

2.196.1 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" />
```

2.197 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
RadialScales

Attributes
RadialScale.Name

Child elements
RadialScale.Radius
RadialScale.StartAngle
RadialScale.SweepAngle
RadialScale.ActionInfo
RadialScale.CustomLabels
RadialScale.GaugeMajorTickMarks
RadialScale.GaugeMinorTickMarks
RadialScale.GaugePointers
RadialScale.Hidden
RadialScale.Interval
RadialScale.IntervalOffset
RadialScale.Logarithmic
RadialScale.LogarithmicBase
RadialScale.MaximumPin
RadialScale.MaximumValue
RadialScale.MinimumPin
RadialScale.MinimumValue
RadialScale.Multiplier
RadialScale.Reversed
RadialScale.ScaleLabels
RadialScale.ScaleRanges
RadialScale.Style

Child elements
RadialScale.TickMarksOnTop
RadialScale.ToolTip
RadialScale.Width

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: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>
```

2.197.1 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
RadialScale

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.197.2 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
RadialScale

The following is the XML Schema definition of the **RadialScale.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.197.3 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" />
```

2.197.4 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" />
```

2.197.5 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" />
```

2.197.6 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" />
```

2.197.7 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" />
```

2.197.8 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" />
```

2.197.9 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
RadialScale

The following is the XML Schema definition of the **RadialScale.IntervalOffset** element.

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

2.197.10 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
RadialGauge

The following is the XML Schema definition of the **RadialScale.Logarithmic** element.

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

2.197.11 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.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.LogarithmicBase** element.

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

2.197.12 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.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.MaximumPin** element.

```
<xsd:element name="MaximumPin" type="ScalePinType" minOccurs="0" />
```

2.197.13 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.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.MaximumValue** element.

```
<xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0" />
```

2.197.14 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" />
```

2.197.15 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" />
```

2.197.16 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.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.Multiplier** element.

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

2.197.17 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.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.Reversed** element.

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

2.197.18 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.

Parent elements
RadialScale

The following is the XML Schema definition of the **RadialScale.ScaleLabels** element.

```
<xsd:element name="ScaleLabels" type="ScaleLabelsType" minOccurs="0" />
```

2.197.19 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" />
```

2.197.20 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" />
```

2.197.21 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" />
```

2.197.22 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" />
```

2.197.23 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" />
```

2.197.24 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">
```

2.197.25 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">
```

2.197.26 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">
```

2.198 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

Child elements
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>
```

2.198.1 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.

Parent elements
RadialPointers

The following is the XML Schema definition of the **RadialPointers.RadialPointer** element.

```
<xsd:element name="RadialPointer" type="RadialPointerType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.199 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.

Parent elements
RadialPointers

Attributes
RadialPointer.Name

Child elements
RadialPointer.ActionInfo
RadialPointer.BarStart
RadialPointer.DistanceFromScale
RadialPointer.GaugeInputValue
RadialPointer.Hidden
RadialPointer.MarkerLength
RadialPointer.MarkerStyle
RadialPointer.Placement
RadialPointer.PointerImage
RadialPointer.SnappingEnabled
RadialPointer.SnappingInterval
RadialPointer.Style
RadialPointer.ToolTip
RadialPointer.Width
RadialPointer.NeedleStyle
RadialPointer.PointerCap
RadialPointer.Type

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>
```

2.199.1 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" />
```

2.199.2 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" />
```

2.199.3 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" />
```

2.199.4 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" />
```

2.199.5 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" />
```

2.199.6 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" />
```

2.199.7 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" />
```

2.199.8 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" />
```

2.199.9 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" />
```

2.199.10 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
RadialPointer

The following is the XML Schema definition of the **RadialPointer.PointerImage** element.

```
<xsd:element name="PointerImage" type="PointerImageType" minOccurs="0" />
```

2.199.11 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" />
```

2.199.12 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" />
```

2.199.13 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" />
```

2.199.14 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" />
```

2.199.15 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" />
```

2.199.16 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.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.NeedleStyle** element.

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

2.199.17 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.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.PointerCap** element.

```
<xsd:element name="PointerCap" type="PointerCapType" minOccurs="0">
```

2.199.18 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.

Parent elements
RadialPointer

The following is the XML Schema definition of the **RadialPointer.Type** element.

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

2.200 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
RadialPointer

Child elements
PointerCap.CapImage
PointerCap.CapStyle
PointerCap.Hidden
PointerCap.OnTop
PointerCap.Reflection
PointerCap.Style
PointerCap.Width

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>
```

2.200.1 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">
```

2.200.2 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">
```

2.200.3 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
PointerCap

The following is the XML Schema definition of the **PointerCap.Hidden** element.

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

2.200.4 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">
```

2.200.5 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">
```

2.200.6 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">
```

2.200.7 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">
```

2.201 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.

Parent elements
PointerCap

Child elements
CapImage.MIMEType
CapImage.Source
CapImage.TransparentColor
CapImage.Value
CapImage.HueColor
CapImage.OffsetX
CapImage.OffsetY

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>
```

2.201.1 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" "/>
```

2.201.2 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" />
```

2.201.3 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
CapImage

The following is the XML Schema definition of the **CapImage.TransparentColor** element.

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

2.201.4 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
CapImage

The following is the XML Schema definition of the **CapImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1" "/>
```

2.201.5 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
CapImage

The following is the XML Schema definition of the **CapImage.HueColor** element.

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

2.201.6 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
CapImage

The following is the XML Schema definition of the **CapImage.OffsetX** element.

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

2.201.7 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
CapImage

The following is the XML Schema definition of the **CapImage.OffsetY** element.

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

2.202 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 for the **StateIndicators** element.

Parent elements
GaugePanel

Child elements
StateIndicators.StateIndicator

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>
```

2.202.1 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.

Parent elements
StateIndicators

The following is the XML Schema definition of the **StateIndicators.StateIndicator** element.

```
<xsd:element name="StateIndicator" type="StateIndicatorType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.203 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.

Parent elements
StateIndicators

Attributes
StateIndicator.Name

Child elements
StateIndicator.ActionInfo
StateIndicator.Height
StateIndicator.Hidden
StateIndicator.Left
StateIndicator.MaximumValue
StateIndicator.MinimumValue
StateIndicator.ParentItem
StateIndicator.ScaleFactor
StateIndicator.StateDataElementName
StateIndicator.StateDataElementOutput
StateIndicator.ToolTip
StateIndicator.Top
StateIndicator.TransformationScope
StateIndicator.TransformationType
StateIndicator.Width
StateIndicator.ZIndex
StateIndicator.Angle
StateIndicator.GaugeInputValue
StateIndicator.IndicatorStates
StateIndicator.IndicatorStyle
StateIndicator.ResizeMode
StateIndicator.StateImage
StateIndicator.IndicatorImage
StateIndicator.Style
StateIndicator.Text
StateIndicator.UseFontPercent

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: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>
```

2.203.1 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" />
```

2.203.2 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" />
```

2.203.3 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
StateIndicator

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

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

2.203.4 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
StateIndicator

The following is the XML Schema definition of the **StateIndicator.Hidden** element.

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

2.203.5 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
StateIndicator

The following is the XML Schema definition of the **StateIndicator.Left** element.

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

2.203.6 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
StateIndicator

The following is the XML Schema definition of the **StateIndicator.MaximumValue** element.

```
<xsd:element name="MaximumValue" type="GaugeInputValueType" minOccurs="0">
```

2.203.7 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
StateIndicator

The following is the XML Schema definition of the **StateIndicator.MinimumValue** element.

```
<xsd:element name="MinimumValue" type="GaugeInputValueType" minOccurs="0">
```

2.203.8 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 ([XMLSCHEMA2/2] 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.

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.ParentItem** element.

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

2.203.9 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.

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.ScaleFactor** element.

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

2.203.10 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.

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.StateDataElementName** element.

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

2.203.11 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
StateIndicator

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>
```

2.203.12 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" />
```

2.203.13 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
StateIndicator

The following is the XML Schema definition of the **StateIndicator.Top** element.

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

2.203.14 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
StateIndicator

The following is the XML Schema definition of the **StateIndicator.TransformationScope** element.

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

2.203.15 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.

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.TransformationType** element.

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

2.203.16 StateIndicator.Width

Applies to RDL 2008/01, RDL 2010/01, and RDL 2016/01

The **StateIndicator.Width** element specifies the width of a StateIndicatoras 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.

Parent elements
StateIndicator

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

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

2.203.17 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.

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.ZIndex** element.

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

2.203.18 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" />
```

2.203.19 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" />
```

2.203.20 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.

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.IndicatorStates** element.

```
<xsd:element name="IndicatorStates" type="IndicatorStatesType" minOccurs="0" />
```

2.203.21 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".

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.IndicatorStyle** element.

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

2.203.22 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
StateIndicator

The following is the XML Schema definition of the **StateIndicator.ResizeMode** element.

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

2.203.23 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
StateIndicator

The following is the XML Schema definition of the **StateIndicator.IndicatorImage** element.

```
<xsd:element name="IndicatorImage" type="IndicatorImageType" minOccurs="0" />
```

2.203.24 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.

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.StateImage** element.

```
<xsd:element name="StateImage" type="StateImageType" minOccurs="0" />
```

2.203.25 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.

Parent elements
StateIndicator

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

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

2.203.26 StateIndicator.Text

Applies to RDL 2008/01

The **StateIndicator.Text** element is ignored.

The following is the parent element of the **StateIndicator.Text** element.

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.Text** element.

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

2.203.27 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.

Parent elements
StateIndicator

The following is the XML Schema definition of the **StateIndicator.UseFontPercent** element.

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

2.204 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.

Parent elements
StateIndicator

Child elements
IndicatorStates.IndicatorState

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>
```

2.204.1 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.

Parent elements
IndicatorStates

The following is the XML Schema definition of the **IndicatorStates.IndicatorState** element.

```
<xsd:element name="IndicatorState" type="IndicatorStateType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.205 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.

Parent elements
IndicatorStates

Attributes
IndicatorState.Name

Child elements
IndicatorState.Color
IndicatorState.IndicatorStyle
IndicatorState.ScaleFactor
IndicatorState.Style
IndicatorState.StartValue
IndicatorState.EndValue
IndicatorState.StateImage
IndicatorState.IndicatorImage
IndicatorState.Text

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: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>
```

2.205.1 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
IndicatorState

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.205.2 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
IndicatorState

The following is the XML Schema definition of the **IndicatorState.Color** element.

```
<xsd:element name="Color" type="xsd:string" minOccurs="1" />
```

2.205.3 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
IndicatorState

The following is the XML Schema definition of the **IndicatorState.IndicatorStyle** element.

```
<xsd:element name="IndicatorStyle" type="xsd:string" minOccurs="1" />
```

2.205.4 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
IndicatorState

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

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

2.205.5 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" />
```

2.205.6 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" />
```

2.205.7 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
IndicatorState

The following is the XML Schema definition of the **IndicatorState.IndicatorImage** element.

```
<xsd:element name="IndicatorImage" type="IndicatorImageType" minOccurs="0" />
```

2.205.8 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
StateIndicator

The following is the XML Schema definition of the **IndicatorState.ScaleFactor** element.

```
<xsd:element name="ScaleFactor" type="xsd:string" minOccurs="1" />
```

2.205.9 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
IndicatorState

The following is the XML Schema definition of the **IndicatorState.StateImage** element.

```
<xsd:element name="StateImage" type="StateImageType" minOccurs="0" />
```

2.205.10 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
IndicatorState

The following is the XML Schema definition of the **IndicatorState.Text** element.

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

2.206 StateImage

Applies to RDL 2008/01

The **StateImage** element is ignored. The following are the parent and child elements for the **StateImage** element.

Parent elements
StateIndicator

Child elements
StateImage.MIMEType
StateImage.Source
StateImage.TransparentColor
StateImage.Value
StateImage.HueColor
StateImage.Transparency

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

2.206.1 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
StateImage

The following is the XML Schema definition of the **StateImage.MIMEType** element.

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

2.206.2 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
StateImage

The following is the XML Schema definition of the **StateImage.Source** element.

```
<xsd:element name="Source" type="xsd:string" minOccurs="1">
```

2.206.3 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
StateImage

The following is the XML Schema definition of the **StateImage.TransparentColor** element.

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

2.206.4 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
StateImage

The following is the XML Schema definition of the **StateImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="1">
```

2.206.5 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
StateImage

The following is the XML Schema definition of the **StateImage.HueColor** element.

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

2.206.6 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
StateImage

The following is the XML Schema definition of the **StateImage.Transparency** element.

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

2.207 IndicatorImage

Applies to RDL 2010/01 and RDL 2016/01

The following are the parent and child elements for the **IndicatorImage** element.

Parent elements
StateIndicator
IndicatorState

Child elements
IndicatorImage.MIMEType
IndicatorImage.Source
IndicatorImage.TransparentColor
IndicatorImage.Value
IndicatorImage.HueColor
IndicatorImage.Transparency

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>
```

2.207.1 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
IndicatorImage

The following is the XML Schema definition of the **IndicatorImage.MIMEType** element.

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

2.207.2 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">
```

2.207.3 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">
```

2.207.4 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">
```

2.207.5 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" />
```

2.207.6 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" />
```

2.208 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
ReportItems
CellContents
CustomReportItem.AltReportItem

Attributes
Map.Name

Child elements
Map.ActionInfo
Map.AntiAliasing
Map.Bookmark
Map.CustomProperties
Map.DataElementName
Map.DataElementOutput
Map.DocumentMapLabel
Map.Height
Map.Left
Map.MapBorderSkin
Map.MapColorScale
Map.MapDataRegions
Map.MapDistanceScale

Child elements
Map.MapLayers
Map.MapLegends
Map.MapTitles
Map.MapViewport
Map.MaximumSpatialElementCount
Map.MaximumTotalPointCount
Map.PageBreak
Map.PageName
Map.RepeatWith
Map.ShadowIntensity
Map.Style
Map.TextAntiAliasingQuality
Map.TileLanguage
Map.ToolTip
Map.Top
Map.Visibility
Map.Width
Map.ZIndex

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: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>
```

2.208.1 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
Map

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

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.208.2 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
Map

The following is the XML Schema definition of the **Map.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.208.3 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
Map

The following is the XML Schema definition of the **Map.AntiAliasing** element.

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

2.208.4 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
Map

The following is the XML Schema definition of the **Map.Bookmark** element.

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

2.208.5 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" />
```

2.208.6 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" />
```

2.208.7 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>
```

2.208.8 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
Map

The following is the XML Schema definition of the **Map.DocumentMapLabel** element.

```
<xsd:element name="DocumentMapLabel" type="StringLocIDType" minOccurs="0" />
```

2.208.9 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" />
```

2.208.10 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
Map

The following is the XML Schema definition of the **Map.Left** element.

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

2.208.11 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
Map

The following is the XML Schema definition of the **Map.MapBorderSkin** element.

```
<xsd:element name="MapBorderSkin" type="MapBorderSkinType" minOccurs="0" />
```

2.208.12 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" />
```

2.208.13 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.

Parent elements
Map

The following is the XML Schema definition of the **Map.MapDataRegions** element.

```
<xsd:element name="MapDataRegions" type="MapDataRegionsType" minOccurs="0" />
```

2.208.14 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.

Parent elements
Map

The following is the XML Schema definition of the **Map.MapDistanceScale** element.

```
<xsd:element name="MapDistanceScale" type="MapDistanceScaleType" minOccurs="0" />
```

2.208.15 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.

Parent elements
Map

The following is the XML Schema definition of the **Map.MapLayers** element.

```
<xsd:element name="MapLayers" type="MapLayersType" minOccurs="0" />
```

2.208.16 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" />
```

2.208.17 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
Map

The following is the XML Schema definition of the **Map.MapTitles** element.

```
<xsd:element name="MapTitles" type="MapTitlesType" minOccurs="0" />
```

2.208.18 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
Map

The following is the XML Schema definition of the **Map.MapViewport** element.

```
<xsd:element name="MapViewport" type="MapViewportType" minOccurs="1" />
```

2.208.19 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" />
```

2.208.20 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" />
```

2.208.21 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
Map

The following is the XML Schema definition of the **Map.PageBreak** element.

```
<xsd:element name="PageBreak" type="PageBreakType" minOccurs="0" />
```

2.208.22 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
Map

The following is the XML Schema definition of the **Map.PageName** element.

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

2.208.23 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
Map

The following is the XML Schema definition of the **Map.RepeatWith** element.

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

2.208.24 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
Map

The following is the XML Schema definition of the **Map.ShadowIntensity** element.

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

2.208.25 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
Map

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

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

2.208.26 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
Map

The following is the XML Schema definition of the **Map.TextAntiAliasingQuality** element.

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

2.208.27 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
Map

The following is the XML Schema definition of the **Map.TileLanguage** element.

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

2.208.28 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
Map

The following is the XML Schema definition of the **Map.ToolTip** element.

```
<xsd:element name="ToolTip" type="StringLocIDType" minOccurs="0" />
```

2.208.29 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
Map

The following is the XML Schema definition of the **Map.Top** element.

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

2.208.30 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
Map

The following is the XML Schema definition of the **Map.Visibility** element.

```
<xsd:element name="Visibility" type="VisibilityType" minOccurs="0" />
```

2.208.31 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" />
```

2.208.32 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" />
```

2.209 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
Map

Child elements
MapBorderSkin.MapBorderSkinType
MapBorderSkin.Style

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>
```

2.209.1 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
MapBorderSkin

The following is the XML Schema definition of the **MapBorderSkin.MapBorderSkinType** element.

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

2.209.2 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
MapBorderSkin

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

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

2.210 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
MapColorScale.ActionInfo
MapColorScale.BottomMargin
MapColorScale.ColorBarBorderColor
MapColorScale.DockOutsideViewport
MapColorScale.Hidden
MapColorScale.HideEndLabels
MapColorScale.LabelBehavior
MapColorScale.LabelFormat
MapColorScale.LabelInterval
MapColorScale.LabelPlacement
MapColorScale.LeftMargin
MapColorScale.MapColorScaleTitle
MapColorScale.MapLocation
MapColorScale.MapSize
MapColorScale.NoDataText
MapColorScale.Position
MapColorScale.RangeGapColor
MapColorScale.RightMargin
MapColorScale.Style
MapColorScale.TickMarkLength
MapColorScale.ToolTip
MapColorScale.TopMargin
MapColorScale.ZIndex

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" />
```

```
      <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>
```

2.210.1 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
MapColorScale

The following is the XML Schema definition of the **MapColorScale.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.210.2 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" />
```

2.210.3 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" />
```

2.210.4 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" />
```

2.210.5 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" />
```

2.210.6 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" />
```

2.210.7 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" />
```

2.210.8 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" />
```

2.210.9 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" />
```

2.210.10 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" />
```

2.210.11 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" />
```

2.210.12 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" />
```

2.210.13 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" />
```

2.210.14 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" />
```

2.210.15 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" />
```

2.210.16 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" />
```

2.210.17 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" />
```

2.210.18 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" />
```

2.210.19 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" />
```

2.210.20 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" />
```

2.210.21 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" />
```

2.210.22 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" />
```

2.210.23 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
MapColorScale

The following is the XML Schema definition of the **MapColorScale.ZIndex** element.

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

2.211 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
MapColorScale

Child elements
MapColorScaleTitle.Caption
MapColorScaleTitle.Style

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>
```

2.211.1 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
MapColorScaleTitle

The following is the XML Schema definition of the **MapColorScaleTitle.Caption** element.

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

2.211.2 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" />
```

2.212 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
MapLocation.Left
MapLocation.Top

Child elements
MapLocation.Unit

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>
```

2.212.1 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.

Parent elements
MapLocation

The following is the XML Schema definition of the **MapLocation.Left** element.

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

2.212.2 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.

Parent elements
MapLocation

The following is the XML Schema definition of the **MapLocation.Top** element.

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

2.212.3 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
MapLocation

The following is the XML Schema definition of the **MapLocation.Unit** element.

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

2.213 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

Child elements
MapSize.Height
MapSize.Unit
MapSize.Width

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>
```

2.213.1 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.

Parent elements
MapSize

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

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

2.213.2 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
MapSize

The following is the XML Schema definition of the **MapSize.Unit** element.

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

2.213.3 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
MapSize

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

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

2.214 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
Map

Child elements
MapDataRegions.MapDataRegion

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>
```

2.214.1 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.

Parent elements
MapDataRegions

The following is the XML Schema definition of the **MapDataRegions.MapDataRegion** element.

```
<xsd:element name="MapDataRegion" type="MapDataRegionType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.215 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.

Parent elements
MapDataRegions

Attributes
MapDataRegion.Name

Child elements
MapDataRegion.DataSetName
MapDataRegion.Filters
MapDataRegion.MapMember

Applies to RDL 2011/01

Child elements
MapDataRegion.Relationship

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>
```

2.215.1 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" />
```

2.215.2 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
MapDataRegion

The following is the XML Schema definition of the **MapDataRegion.DataSetName** element.

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

2.215.3 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
MapDataRegion

The following is the XML Schema definition of the **MapDataRegion.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

2.215.4 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
MapDataRegion

The following is the XML Schema definition of the **MapDataRegion.MapMember** element.

```
<xsd:element name="MapMember" type="MapMemberType" minOccurs="0" />
```

2.215.5 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
MapDataRegion

The following is the XML Schema definition of the **MapDataRegion.Relationship** element.

```
<xsd:element name ="Relationship" type="RelationshipType" minOccurs="0" />
```

2.216 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
MapDataRegion
MapMember.MapMember

Child elements
MapMember.Group
MapMember.MapMember

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

2.216.1 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
MapMember

The following is the XML Schema definition of the **MapMember.Group** element.

```
<xsd:element name="Group" type="GroupType" minOccurs="1" />
```

2.216.2 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
MapMember

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

```
<xsd:element name="MapMember" type="MapMemberType" minOccurs="0" />
```

2.217 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
Map

Child elements
MapDistanceScale.ActionInfo
MapDistanceScale.BottomMargin
MapDistanceScale.DockOutsideViewport
MapDistanceScale.Hidden
MapDistanceScale.LeftMargin
MapDistanceScale.MapLocation
MapDistanceScale.MapSize
MapDistanceScale.Position
MapDistanceScale.RightMargin
MapDistanceScale.ScaleBorderColor
MapDistanceScale.ScaleColor
MapDistanceScale.Style
MapDistanceScale.ToolTip
MapDistanceScale.TopMargin
MapDistanceScale.ZIndex

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>
```

2.217.1 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" />
```

2.217.2 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" />
```

2.217.3 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 ([XMLSCHEMA2/2] 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.

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" />
```

2.217.4 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" />
```

2.217.5 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" />
```

2.217.6 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" />
```

2.217.7 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" />
```

2.217.8 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" />
```

2.217.9 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" />
```

2.217.10 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" />
```

2.217.11 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" />
```

2.217.12 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" />
```

2.217.13 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" />
```

2.217.14 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
MapDistanceScale

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

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

2.217.15 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
MapDistanceScale

The following is the XML Schema definition of the **MapDistanceScale.ZIndex** element.

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

2.218 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
Map

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>
```

2.218.1 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
MapLayers

The following is the XML Schema definition of the **MapLayers.MapLineLayer** element.

```
<xsd:element name="MapLineLayer" type="MapLineLayerType" />
```

2.218.2 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
MapLayers

The following is the XML Schema definition of the **MapLayers.MapPointLayer** element.

```
<xsd:element name="MapPointLayer" type="MapPointLayerType" />
```

2.218.3 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
MapLayers

The following is the XML Schema definition of the **MapLayers.MapPolygonLayer** element.

```
<xsd:element name="MapPolygonLayer" type="MapPolygonLayerType" />
```

2.218.4 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
MapLayers

The following is the XML Schema definition of the **MapLayers.MapTileLayer** element.

```
<xsd:element name="MapTileLayer" type="MapTileLayerType" />
```

2.219 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
MapLayers

Attributes
MapLineLayer.Name

Child elements
MapLineLayer.DataElementName
MapLineLayer.DataElementOutput
MapLineLayer.MapBindingFieldPairs
MapLineLayer.MapDataRegionName
MapLineLayer.MapFieldDefinitions
MapLineLayer.MapLineRules
MapLineLayer.MapLines
MapLineLayer.MapLineTemplate
MapLineLayer.MapShapefile
MapLineLayer.MapSpatialDataSet
MapLineLayer.MapSpatialDataRegion
MapLineLayer.MaximumZoom
MapLineLayer.MinimumZoom
MapLineLayer.Transparency
MapLineLayer.VisibilityMode

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>
      <!--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>
```

2.219.1 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" />
```

2.219.2 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" />
```

2.219.3 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
MapLineLayer

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>
```

2.219.4 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
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.MapBindingFieldPairs** element.

```
<xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType" minOccurs="0" />
```

2.219.5 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
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.MapDataRegionName** element.

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

2.219.6 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
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.MapFieldDefinitions** element.

```
<xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType" minOccurs="0" />
```

2.219.7 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
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.MapLineRules** element.

```
<xsd:element name="MapLineRules" type="MapLineRulesType" minOccurs="0" />
```

2.219.8 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
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.MapLines** element.

```
<xsd:element name="MapLines" type="MapLinesType" minOccurs="0" />
```

2.219.9 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
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.MapLineTemplate** element.

```
<xsd:element name="MapLineTemplate" type="MapLineTemplateType" minOccurs="0" />
```

2.219.10 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
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.MapSpatialDataRegion** element.

```
<xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType" minOccurs="0" />
```

2.219.11 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
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.MapShapefile** element.

```
<xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
```

2.219.12 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
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.MapSpatialDataSet** element.

```
<xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType" minOccurs="0" />
```

2.219.13 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
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.MaximumZoom** element.

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

2.219.14 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
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.MinimumZoom** element.

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

2.219.15 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.

Parent elements
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.Transparency** element.

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

2.219.16 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.

Parent elements
MapLineLayer

The following is the XML Schema definition of the **MapLineLayer.VisibilityMode** element.

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

2.220 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.

Parent elements
MapLineLayer
MapPointLayer

Parent elements
MapPolygonLayer
MapElementView

Child elements
MapBindingFieldPairs.MapBindingFieldPair

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>
```

2.220.1 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.

Parent elements
MapBindingFieldPairs

The following is the XML Schema definition of the **MapBindingFieldPairs.MapBindingFieldPair** element.

```
<xsd:element name="MapBindingFieldPair" type="MapBindingFieldPairType"
             minOccurs="1" maxOccurs="unbounded" />
```

2.221 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 MapElementView, 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
MapBindingFieldPairs

Child elements
MapBindingFieldPair.BindingExpression
MapBindingFieldPair.FieldName

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>
```

2.221.1 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
MapBindingFieldPair

The following is the XML Schema definition of the **MapBindingFieldPair.BindingExpression** element.

```
<xsd:element name="BindingExpression" type="xsd:string" minOccurs="1" />
```

2.221.2 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
MapBindingFieldPair

The following is the XML Schema definition of the **MapBindingFieldPair.FieldName** element.

```
<xsd:element name="FieldName" type="xsd:string" minOccurs="1" />
```

2.222 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
MapFieldDefinitions.MapFieldDefinition

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>
```

2.222.1 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
MapFieldDefinitions

The following is the XML Schema definition of the **MapFieldDefinitions.MapFieldDefinition** element.

```
<xsd:element name="MapFieldDefinition" type="MapFieldDefinitionType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.223 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
MapFieldDefinitions

Child elements
MapFieldDefinition.DataType
MapFieldDefinition.Name

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>
```

2.223.1 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
MapFieldDefinition

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>
```

2.223.2 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" />
```

2.224 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
MapLineLayer

Child elements
MapLineRules.MapColorPaletteRule
MapLineRules.MapColorRangeRule
MapLineRules.MapCustomColorRule
MapLineRules.MapSizeRule

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>
```

2.224.1 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.

Parent elements
MapLineRules

The following is the XML Schema definition of the **MapLineRules.MapColorPaletteRule** element.

```
<xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType"
             minOccurs="0" />
```

2.224.2 MapLineRules.MapColorRangeRule

Applies to RDL 2010/01 and RDL 2016/01

The **MapLineRules.MapColorRangeRule** element specifies a color range to be used for colorizing 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.

Parent elements
MapLineRules

The following is the XML Schema definition of the **MapLineRules.MapColorRangeRule** element.

```
<xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType" minOccurs="0" />
```

2.224.3 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.

Parent elements
MapLineRules

The following is the XML Schema definition of the **MapLineRules.MapCustomColorRule** element.

```
<xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType" minOccurs="0" />
```

2.224.4 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.

Parent elements
MapLineRules

The following is the XML Schema definition of the **MapLineRules.MapSizeRule** element.

```
<xsd:element name="MapSizeRule" type="MapSizeRuleType" minOccurs="0" />
```

2.225 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.

Parent elements
MapLineRules
MapPointRules
MapPolygonRules

Child elements
MapColorPaletteRule.BucketCount
MapColorPaletteRule.DataElementName
MapColorPaletteRule.DataElementOutput

Child elements
MapColorPaletteRule.DataValue
MapColorPaletteRule.DistributionType
MapColorPaletteRule.EndValue
MapColorPaletteRule.LegendName
MapColorPaletteRule.LegendText
MapColorPaletteRule.MapBuckets
MapColorPaletteRule.Palette
MapColorPaletteRule.ShowInColorScale
MapColorPaletteRule.StartValue

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>
```

2.225.1 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
MapColorPaletteRule

The following is the XML Schema definition of the **MapColorPaletteRule.BucketCount** element.

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

2.225.2 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
MapColorPaletteRule

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

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

2.225.3 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
MapColorPaletteRule

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>
```

2.225.4 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
MapColorPaletteRule

The following is the XML Schema definition of the **MapColorPaletteRule.DataValue** element.

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

2.225.5 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
MapColorPaletteRule

The following is the XML Schema definition of the **MapColorPaletteRule.DistributionType** element.

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

2.225.6 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
MapColorPaletteRule

The following is the XML Schema definition of the **MapColorPaletteRule.EndValue** element.

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

2.225.7 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
MapColorPaletteRule

The following is the XML Schema definition of the **MapColorPaletteRule.LegendName** element.

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

2.225.8 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
MapColorPaletteRule

The following is the XML Schema definition of the **MapColorPaletteRule.LegendText** element.

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

2.225.9 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
MapColorPaletteRule

The following is the XML Schema definition of the **MapColorPaletteRule.MapBuckets** element.

```
<xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
```

2.225.10 MapColorPaletteRule.Palette

The **MapColorPaletteRule.Palette** element specifies the palette to be used for colorizing 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
MapColorPaletteRule

The following is the XML Schema definition of the **MapColorPaletteRule.Palette** element.

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

2.225.11 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
MapColorPaletteRule

The following is the XML Schema definition of the **MapColorPaletteRule.ShowInColorScale** element.

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

2.225.12 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
MapColorPaletteRule

The following is the XML Schema definition of the **MapColorPaletteRule.StartValue** element.

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

2.226 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
MapColorPaletteRule
MapColorRangeRule
MapCustomColorRule
MapMarkerRule
MapSizeRule

Child elements
MapBuckets.MapBucket

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>
```

2.226.1 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
MapBuckets

The following is the XML Schema definition of the **MapBuckets.MapBucket** element.

```
<xsd:element name="MapBucket" type="MapBucketType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.227 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
MapBuckets

Child elements
MapBucket.EndValue
MapBucket.StartValue

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>
```

2.227.1 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
MapBucket

The following is the XML Schema definition of the **MapBucket.EndValue** element.

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

2.227.2 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" />
```

2.228 MapColorRangeRule

Applies to RDL 2010/01 and RDL 2016/01

The **MapColorRangeRule** element specifies a color range to be used for colorizing **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
MapLineRules
MapPointRules
MapPolygonRules

Child elements
MapColorRangeRule.BucketCount
MapColorRangeRule.DataElementName
MapColorRangeRule.DataElementOutput
MapColorRangeRule.DataValue

Child elements
MapColorRangeRule.DistributionType
MapColorRangeRule.EndColor
MapColorRangeRule.EndValue
MapColorRangeRule.LegendName
MapColorRangeRule.LegendText
MapColorRangeRule.MapBuckets
MapColorRangeRule.MiddleColor
MapColorRangeRule.ShowInColorScale
MapColorRangeRule.StartColor
MapColorRangeRule.StartValue

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>
```

2.228.1 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
MapColorRangeRule

The following is the XML Schema definition of the **MapColorRangeRule.BucketCount** element.

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

2.228.2 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
MapColorRangeRule

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

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

2.228.3 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.

Parent elements
MapColorRangeRule

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>
```

2.228.4 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.

Parent elements
MapColorRangeRule

The following is the XML Schema definition of the **MapColorRangeRule.DataValue** element.

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

2.228.5 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" />
```

2.228.6 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" />
```

2.228.7 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" />
```

2.228.8 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
MapColorRangeRule

The following is the XML Schema definition of the **MapColorRangeRule.LegendName** element.

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

2.228.9 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
MapColorRangeRule

The following is the XML Schema definition of the **MapColorRangeRule.LegendText** element.

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

2.228.10 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.

Parent elements
MapColorRangeRule

The following is the XML Schema definition of the **MapColorRangeRule.MapBuckets** element.

```
<xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
```

2.228.11 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.

Parent elements
MapColorRangeRule

The following is the XML Schema definition of the **MapColorRangeRule.MiddleColor** element.

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

2.228.12 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.

Parent elements
MapColorRangeRule

The following is the XML Schema definition of the **MapColorRangeRule.ShowInColorScale** element.

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

2.228.13 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
MapColorRangeRule

The following is the XML Schema definition of the **MapColorRangeRule.StartColor** element.

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

2.228.14 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
MapColorRangeRule

The following is the XML Schema definition of the **MapColorRangeRule.StartValue** element.

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

2.229 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
MapLineRules
MapPointRules
MapPolygonRules

Child elements
MapCustomColorRule.BucketCount
MapCustomColorRule.DataElementName
MapCustomColorRule.DataElementOutput
MapCustomColorRule.DataValue
MapCustomColorRule.DistributionType
MapCustomColorRule.EndValue
MapCustomColorRule.LegendName
MapCustomColorRule.LegendText
MapCustomColorRule.MapBuckets
MapCustomColorRule.MapCustomColors
MapCustomColorRule.ShowInColorScale
MapCustomColorRule.StartValue

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>
```

2.229.1 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
MapCustomColorRule

The following is the XML Schema definition of the **MapCustomColorRule.BucketCount** element.

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

2.229.2 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
MapCustomColorRule

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

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

2.229.3 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.

Parent elements
MapCustomColorRule

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>
```

2.229.4 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.

Parent elements
MapCustomColorRule

The following is the XML Schema definition of the **MapCustomColorRule.DataValue** element.

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

2.229.5 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" />
```

2.229.6 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" />
```

2.229.7 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
MapCustomColorRule

The following is the XML Schema definition of the **MapCustomColorRule.LegendName** element.

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

2.229.8 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
MapCustomColorRule

The following is the XML Schema definition of the **MapCustomColorRule.LegendText** element.

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

2.229.9 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
MapCustomColorRule

The following is the XML Schema definition of the **MapCustomColorRule.MapBuckets** element.

```
<xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
```

2.229.10 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.

Parent elements
MapCustomColorRule

The following is the XML Schema definition of the **MapCustomColorRule.MapCustomColors** element.

```
<xsd:element name="MapCustomColors" type="MapCustomColorsType" minOccurs="1" />
```

2.229.11 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.

Parent elements
MapCustomColorRule

The following is the XML Schema definition of the **MapColorRangeRule.ShowInColorScale** element.

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

2.229.12 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.

Parent elements
MapCustomColorRule

The following is the XML Schema definition of the **MapCustomColorRule.StartValue** element.

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

2.230 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
MapCustomColorRule

Child elements
MapCustomColors.MapCustomColor

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>
```

2.230.1 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
MapCustomColors

The following is the XML Schema definition of the **MapCustomColors.MapCustomColor** element.

```
<xsd:element name="MapCustomColor" type="xsd:string" minOccurs="1"
             maxOccurs="unbounded" />
```

2.231 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
MapLineRules
MapPointRules

Child elements
MapSizeRule.BucketCount
MapSizeRule.DataElementName
MapSizeRule.DataElementOutput
MapSizeRule.DataValue
MapSizeRule.DistributionType
MapSizeRule.EndSize
MapSizeRule.EndValue
MapSizeRule.LegendName
MapSizeRule.LegendText
MapSizeRule.MapBuckets
MapSizeRule.StartSize
MapSizeRule.StartValue

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>
```

2.231.1 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
MapSizeRule

The following is the XML Schema definition of the **MapSizeRule.BucketCount** element.

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

2.231.2 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
MapSizeRule

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

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

2.231.3 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
MapSizeRule

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>
```

2.231.4 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
MapSizeRule

The following is the XML Schema definition of the **MapSizeRule.DataValue** element.

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

2.231.5 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" />
```

2.231.6 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
MapSizeRule

The following is the XML Schema definition of the **MapSizeRule.EndSize** element.

```
<xsd:element name="EndSize" type="xsd:string" minOccurs="1" />
```

2.231.7 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" />
```

2.231.8 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
MapSizeRule

The following is the XML Schema definition of the **MapSizeRule.LegendName** element.

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

2.231.9 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
MapSizeRule

The following is the XML Schema definition of the **MapSizeRule.LegendText** element.

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

2.231.10 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.

Parent elements
MapSizeRule

The following is the XML Schema definition of the **MapSizeRule.MapBuckets** element.

```
<xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
```

2.231.11 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.

Parent elements
MapSizeRule

The following is the XML Schema definition of the **MapSizeRule.StartSize** element.

```
<xsd:element name="StartSize" type="xsd:string" minOccurs="1" />
```

2.231.12 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.

Parent elements
MapSizeRule

The following is the XML Schema definition of the **MapSizeRule.StartValue** element.

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

2.232 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
MapLineLayer

Child elements
MapLines.MapLine

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>
```

2.232.1 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
MapLines

The following is the XML Schema definition of the **MapLines.MapLine** element.

```
<xsd:element name="MapLine" type="MapLineType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.233 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.

Parent elements
MapLines

Child elements
MapLine.MapFields
MapLine.MapLineTemplate
MapLine.UseCustomLineTemplate
MapLine.VectorData

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>
```

2.233.1 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.

Parent elements
MapLine

The following is the XML Schema definition of the **MapLine.MapFields** element.

```
<xsd:element name="MapFields" type="MapFieldsType" minOccurs="0" />
```

2.233.2 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
MapLine

The following is the XML Schema definition of the **MapLine.MapLineTemplate** element.

```
<xsd:element name="MapLineTemplate" type="MapLineTemplateType" minOccurs="0" />
```

2.233.3 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
MapLine

The following is the XML Schema definition of the **MapLine.UseCustomLineTemplate** element.

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

2.233.4 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
MapLine

The following is the XML Schema definition of the **MapLine.VectorData** element.

```
<xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
```

2.234 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
MapLine
MapPoint
MapPolygon

Child elements
MapFields.MapField

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>
```

2.234.1 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
MapFields

The following is the XML Schema definition of the **MapFields.MapField** element.

```
<xsd:element name="MapField" type="MapFieldType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.235 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
MapFields

Child elements
MapField.Name
MapField.Value

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>
```

2.235.1 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
MapField

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

```
<xsd:element name="Name" type="xsd:string" minOccurs="1" />
```

2.235.2 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" />
```

2.236 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
MapLine
MapLineLayer

Child elements
MapLineTemplate.ActionInfo
MapLineTemplate.DataElementLabel
MapLineTemplate.DataElementName
MapLineTemplate.DataElementOutput
MapLineTemplate.Hidden
MapLineTemplate.Label
MapLineTemplate.LabelPlacement
MapLineTemplate.OffsetX
MapLineTemplate.OffsetY
MapLineTemplate.Style
MapLineTemplate.ToolTip
MapLineTemplate.Width

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" />
```

```
    <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>
```

2.236.1 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.

Parent elements
MapLineTemplate

The following is the XML Schema definition of the **MapLineTemplate.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.236.2 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.

Parent elements
MapLineTemplate

The following is the XML Schema definition of the **MapLineTemplate.DataElementLabel** element.

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

2.236.3 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" />
```

2.236.4 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>
```

2.236.5 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
MapLineTemplate

The following is the XML Schema definition of the **MapLineTemplate.Hidden** element.

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

2.236.6 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
MapLineTemplate

The following is the XML Schema definition of the **MapLineTemplate.Label** element.

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

2.236.7 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
MapLineTemplate

The following is the XML Schema definition of the **MapLineTemplate.LabelPlacement** element.

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

2.236.8 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" />
```

2.236.9 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" />
```

2.236.10 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" />
```

2.236.11 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" />
```

2.236.12 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" />
```

2.237 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
MapLineLayer
MapPointLayer
MapPolygonLayer

Child elements
MapShapefile.MapFieldNames
MapShapefile.Source

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>
```

2.237.1 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
MapShapefile

The following is the XML Schema definition of the **MapShapefile.MapFieldNames** element.

```
<xsd:element name="MapFieldNames" type="MapFieldNamesType" minOccurs="0" />
```

2.237.2 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.

Parent elements
MapShapefile

The following is the XML Schema definition of the **MapShapefile.Source** element.

```
<xsd:element name="Source" type="xsd:string" minOccurs="1" />
```

2.238 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.

Parent elements
MapShapefile
MapSpatialDataSet

Child elements
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>
```

2.238.1 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.

Parent elements
MapFieldNames

The following is the XML Schema definition of the **MapFieldNames.MapFieldName** element.

```
<xsd:element name="MapFieldName" type="xsd:string" minOccurs="1"
             maxOccurs="unbounded" />
```

2.239 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
MapLineLayer
MapPointLayer
MapPolygonLayer

Child elements
MapSpatialDataRegion.VectorData

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>
```

2.239.1 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
MapSpatialDataRegion

The following is the XML Schema definition of the **MapSpatialDataRegion.VectorData** element.

```
<xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
```

2.240 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
MapLineLayer
MapPointLayer
MapPolygonLayer

Child elements
MapSpatialDataSet.DataSetName
MapSpatialDataSet.MapFieldNames
MapSpatialDataSet.SpatialField

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>
```

2.240.1 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
MapSpatialDataSet

The following is the XML Schema definition of the **MapSpatialDataSet.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" minOccurs="1" />
```

2.240.2 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
MapSpatialDataSet

The following is the XML Schema definition of the **MapSpatialDataSet.MapFieldNames** element.

```
<xsd:element name="MapFieldNames" type="MapFieldNamesType" minOccurs="0" />
```

2.240.3 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
MapSpatialDataSet

The following is the XML Schema definition of the **MapSpatialDataSet.SpatialField** element.

```
<xsd:element name="SpatialField" type="xsd:string" minOccurs="1" />
```

2.241 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
MapLayers

Attributes
MapPointLayer.Name

Child elements
MapPointLayer.DataElementName

Child elements
MapPointLayer.DataElementOutput
MapPointLayer.MapBindingFieldPairs
MapPointLayer.MapDataRegionName
MapPointLayer.MapFieldDefinitions
MapPointLayer.MapPointRules
MapPointLayer.MapPoints
MapPointLayer.MapMarkerTemplate
MapPointLayer.MapShapefile
MapPointLayer.MapSpatialDataSet
MapPointLayer.MapSpatialDataRegion
MapPointLayer.MaximumZoom
MapPointLayer.MinimumZoom
MapPointLayer.Transparency
MapPointLayer.VisibilityMode

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>
    <xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
    <xsd:anyAttribute namespace="##other" processContents="lax" />
  </xsd:complexType>
```

2.241.1 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" />
```

2.241.2 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" />
```

2.241.3 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
MapPointLayer

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>
```

2.241.4 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
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.MapBindingFieldPairs** element.

```
<xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType" minOccurs="0" />
```

2.241.5 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
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.MapDataRegionName** element.

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

2.241.6 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
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.MapFieldDefinitions** element.

```
<xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType" minOccurs="0" />
```

2.241.7 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
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.MapPointRules** element.

```
<xsd:element name="MapPointRules" type="MapPointRulesType" minOccurs="0" />
```

2.241.8 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.

Parent elements
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.MapPoints** element.

```
<xsd:element name="MapPoints" type="MapPointsType" minOccurs="0" />
```

2.241.9 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.

Parent elements
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.MapMarkerTemplate** element.

```
<xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType" minOccurs="0" />
```

2.241.10 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.

Parent elements
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.MapSpatialDataRegion** element.

```
<xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType" minOccurs="0" />
```

2.241.11 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
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.MapShapefile** element.

```
<xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
```

2.241.12 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
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.MapSpatialDataSet** element.

```
<xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType" minOccurs="0" />
```

2.241.13 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
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.MaximumZoom** element.

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

2.241.14 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
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.MinimumZoom** element.

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

2.241.15 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
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.Transparency** element.

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

2.241.16 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
MapPointLayer

The following is the XML Schema definition of the **MapPointLayer.VisibilityMode** element.

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

2.242 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
MapMarkerTemplate.ActionInfo
MapMarkerTemplate.DataElementLabel
MapMarkerTemplate.DataElementName

Child elements
MapMarkerTemplate.DataElementOutput
MapMarkerTemplate.Hidden
MapMarkerTemplate.Label
MapMarkerTemplate.LabelPlacement
MapMarkerTemplate.MapMarker
MapMarkerTemplate.OffsetX
MapMarkerTemplate.OffsetY
MapMarkerTemplate.Size
MapMarkerTemplate.Style
MapMarkerTemplate.ToolTip

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>
```

2.242.1 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.

Parent elements
MapMarkerTemplate

The following is the XML Schema definition of the **MapMarkerTemplate.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.242.2 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.

Parent elements
MapMarkerTemplate

The following is the XML Schema definition of the **MapMarkerTemplate.DataElementLabel** element.

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

2.242.3 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.

Parent elements
MapMarkerTemplate

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

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

2.242.4 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
MapMarkerTemplate

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>
```

2.242.5 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 ([XMLSCHEMA2/2] 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 **MapMarkerTemplate.Hidden** element.

Parent elements
MapMarkerTemplate

The following is the XML Schema definition of the **MapMarkerTemplate.Hidden** element.

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

2.242.6 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
MapMarkerTemplate

The following is the XML Schema definition of the **MapMarkerTemplate.Label** element.

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

2.242.7 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
MapMarkerTemplate

The following is the XML Schema definition of the **MapMarkerTemplate.LabelPlacement** element.

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

2.242.8 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.

Parent elements
MapMarkerTemplate

The following is the XML Schema definition of the **MapMarkerTemplate.MapMarker** element.

```
<xsd:element name="MapMarker" type="MapMarkerType" minOccurs="0" />
```

2.242.9 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.

Parent elements
MapMarkerTemplate

The following is the XML Schema definition of the **MapMarkerTemplate.OffsetX** element.

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

2.242.10 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.

Parent elements
MapMarkerTemplate

The following is the XML Schema definition of the **MapMarkerTemplate.OffsetY** element.

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

2.242.11 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
MapMarkerTemplate

The following is the XML Schema definition of the **MapMarkerTemplate.Size** element.

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

2.242.12 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
MapMarkerTemplate

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

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

2.242.13 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
MapMarkerTemplate

The following is the XML Schema definition of the **MapMarkerTemplate.ToolTip** element.

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

2.243 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
MapPointLayer
MapPolygonLayer

Child elements
MapLineRules.MapColorPaletteRule
MapLineRules.MapColorRangeRule
MapLineRules.MapCustomColorRule
MapPointRules.MapMarkerRule
MapLineRules.MapSizeRule

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>
```

2.243.1 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
MapPointRules

The following is the XML Schema definition of the **MapPointRules.MapColorPaletteRule** element.

```
<xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType" minOccurs="0" />
```

2.243.2 MapPointRules.MapColorRangeRule

Applies to RDL 2010/01 and RDL 2016/01

The **MapPointRules.MapColorRangeRule** element specifies a color range to be used for colorizing 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
MapPointRules

The following is the XML Schema definition of the **MapPointRules.MapColorRangeRule** element.

```
<xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType" minOccurs="0" />
```

2.243.3 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
MapPointRules

The following is the XML Schema definition of the **MapPointRules.MapCustomColorRule** element.

```
<xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType" minOccurs="0" />
```

2.243.4 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
MapPointRules

The following is the XML Schema definition of the **MapPointRules.MapMarkerRule** element.

```
<xsd:element name="MapMarkerRule" type="MapMarkerRuleType" minOccurs="0" />
```

2.243.5 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.

Parent elements
MapPointRules

The following is the XML Schema definition of the **MapPointRules.MapSizeRule** element.

```
<xsd:element name="MapSizeRule" type="MapSizeRuleType" minOccurs="0" />
```

2.244 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.

Parent elements
MapPointRules

Child elements
MapMarkerRule.BucketCount
MapMarkerRule.DataElementName
MapMarkerRule.DataElementOutput
MapMarkerRule.DataValue
MapMarkerRule.DistributionType
MapMarkerRule.EndValue
MapMarkerRule.LegendName
MapMarkerRule.LegendText
MapMarkerRule.MapBuckets
MapMarkerRule.MapMarkers
MapMarkerRule.StartValue

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" />
```

```
    <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>
```

2.244.1 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
MapMarkerRule

The following is the XML Schema definition of the **MapMarkerRule.BucketCount** element.

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

2.244.2 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
MapMarkerRule

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

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

2.244.3 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
MapMarkerRule

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>
```

2.244.4 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
MapMarkerRule

The following is the XML Schema definition of the **MapMarkerRule.DataValue** element.

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

2.244.5 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" />
```

2.244.6 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" />
```

2.244.7 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
MapMarkerRule

The following is the XML Schema definition of the **MapMarkerRule.LegendName** element.

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

2.244.8 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
MapMarkerRule

The following is the XML Schema definition of the **MapMarkerRule.LegendText** element.

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

2.244.9 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
MapMarkerRule

The following is the XML Schema definition of the **MapMarkerRule.MapBuckets** element.

```
<xsd:element name="MapBuckets" type="MapBucketsType" minOccurs="0" />
```

2.244.10 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
MapMarkerRule

The following is the XML Schema definition of the **MapMarkerRule.MapMarkers** element.

```
<xsd:element name="MapMarkers" type="MapMarkersType" minOccurs="1" />
```

2.244.11 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" />
```

2.245 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.

Parent elements
MapMarkerRule

Child elements
MapMarkers.MapMarker

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>
```

2.245.1 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.

Parent elements
MapMarkers

The following is the XML Schema definition of the **MapMarkers.MapMarker** element.

```
<xsd:element name="MapMarker" type="MapMarkerType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.246 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.

Parent elements
MapMarkerTemplate
MapMarkers

Child elements
MapMarker.MapMarkerImage
MapMarker.MapMarkerStyle

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>
```

2.246.1 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.

Parent elements
MapMarker

The following is the XML Schema definition of the **MapMarker.MapMarkerImage** element.

```
<xsd:element name="MapMarkerImage" type="MapMarkerImageType" minOccurs="0" />
```

2.246.2 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" />
```

2.247 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
MapMarkerImage.MIMEType
MapMarkerImage.ResizeMode
MapMarkerImage.Source
MapMarkerImage.TransparentColor
MapMarkerImage.Value

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:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

2.247.1 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" />
```

2.247.2 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" />
```

2.247.3 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" />
```

2.247.4 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" />
```

2.247.5 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
MapMarkerImage

The following is the XML Schema definition of the **MapMarkerImage.Value** element.

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

2.248 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
MapPoints.MapPoint

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>
```

2.248.1 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
MapPoints

The following is the XML Schema definition of the **MapPoints.MapPoint** element.

```
<xsd:element name="MapPoint" type="MapPointType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.249 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
MapPoints

Child elements
MapPoint.MapFields
MapPoint.MapMarkerTemplate
MapPoint.UseCustomPointTemplate
MapPoint.VectorData

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>
```

2.249.1 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
MapPoint

The following is the XML Schema definition of the **MapPoint.MapFields** element.

```
<xsd:element name="MapFields" type="MapFieldsType" minOccurs="0" />
```

2.249.2 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
MapPoint

The following is the XML Schema definition of the **MapPoint.MapMarkerTemplate** element.

```
<xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType"
             minOccurs="0" />
```

2.249.3 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
MapPoint

The following is the XML Schema definition of the **MapPoint.UseCustomPointTemplate** element.

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

2.249.4 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
MapPoint

The following is the XML Schema definition of the **MapPoint.VectorData** element.

```
<xsd:element name="VectorData" type="xsd:string" minOccurs="1" />
```

2.250 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
MapLayers

Attributes
MapPolygonLayer.Name

Child elements
MapPolygonLayer.DataElementName
MapPolygonLayer.DataElementOutput
MapPolygonLayer.MapBindingFieldPairs
MapPolygonLayer.MapCenterPointRules
MapPolygonLayer.MapMarkerTemplate
MapPolygonLayer.MapDataRegionName

Child elements
MapPolygonLayer.MapFieldDefinitions
MapPolygonLayer.MapPolygonRules
MapPolygonLayer.MapPolygons
MapPolygonLayer.MapPolygonTemplate
MapPolygonLayer.MapShapefile
MapPolygonLayer.MapSpatialDataSet
MapPolygonLayer.MapSpatialDataRegion
MapPolygonLayer.MaximumZoom
MapPolygonLayer.MinimumZoom
MapPolygonLayer.Transparency
MapPolygonLayer.VisibilityMode

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>
```

2.250.1 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" />
```

2.250.2 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" />
```

2.250.3 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>
```

2.250.4 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" />
```

2.250.5 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.

Parent elements
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.MapCenterPointRules** element.

```
<xsd:element name="MapCenterPointRules" type="MapPointRulesType" minOccurs="0" />
```

2.250.6 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.

Parent elements
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.MapMarkerTemplate** element.

```
<xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType"
             minOccurs="0" />
```

2.250.7 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.

Parent elements
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.MapDataRegionName** element.

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

2.250.8 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
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.MapFieldDefinitions** element.

```
<xsd:element name="MapFieldDefinitions" type="MapFieldDefinitionsType"
             minOccurs="0" />
```

2.250.9 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
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.MapPolygonRules** element.

```
<xsd:element name="MapPolygonRules" type="MapPolygonRulesType" minOccurs="0" />
```

2.250.10 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
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.MapPolygons** element.

```
<xsd:element name="MapPolygons" type="MapPolygonsType" minOccurs="0" />
```

2.250.11 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
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.MapPolygonTemplate** element.

```
<xsd:element name="MapPolygonTemplate" type="MapPolygonTemplateType"
             minOccurs="0" />
```

2.250.12 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
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.MapSpatialDataRegion** element.

```
<xsd:element name="MapSpatialDataRegion" type="MapSpatialDataRegionType"
             minOccurs="0" />
```

2.250.13 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
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.MapShapefile** element.

```
<xsd:element name="MapShapefile" type="MapShapefileType" minOccurs="0" />
```

2.250.14 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
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.MapSpatialDataSet** element.

```
<xsd:element name="MapSpatialDataSet" type="MapSpatialDataSetType"
             minOccurs="0" />
```

2.250.15 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
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.MaximumZoom** element.

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

2.250.16 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
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.MinimumZoom** element.

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

2.250.17 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.

Parent elements
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.Transparency** element.

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

2.250.18 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.

Parent elements
MapPolygonLayer

The following is the XML Schema definition of the **MapPolygonLayer.VisibilityMode** element.

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

2.251 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.

Parent elements
MapPolygonLayer

Child elements
MapLineRules.MapColorPaletteRule
MapLineRules.MapColorRangeRule
MapLineRules.MapCustomColorRule

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>
```

2.251.1 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
MapPolygonRules

The following is the XML Schema definition of the **MapPolygonRules.MapColorPaletteRule** element.

```
<xsd:element name="MapColorPaletteRule" type="MapColorPaletteRuleType"
             minOccurs="0" />
```

2.251.2 MapPolygonRules.MapColorRangeRule

Applies to RDL 2010/01 and RDL 2016/01

The **MapPolygonRules.MapColorRangeRule** element specifies a color range to be used for colorizing 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
MapPolygonRules

The following is the XML Schema definition of the **MapPolygonRules.MapColorRangeRule** element.

```
<xsd:element name="MapColorRangeRule" type="MapColorRangeRuleType"
             minOccurs="0" />
```

2.251.3 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
MapPolygonRules

The following is the XML Schema definition of the **MapPolygonRules.MapCustomColorRule** element.

```
<xsd:element name="MapCustomColorRule" type="MapCustomColorRuleType"
             minOccurs="0" />
```

2.252 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
MapPolygonLayer

Child elements
MapPolygons.MapPolygon

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>
```

2.252.1 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
MapPolygons

The following is the XML Schema definition of the **MapPolygons.MapPolygon** element.

```
<xsd:element name="MapPolygon" type="MapPolygonType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.253 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.

Parent elements
MapPolygons

Child elements
MapPolygon.MapMarkerTemplate
MapPolygon.MapFields
MapPolygon.MapPolygonTemplate
MapPolygon.UseCustomCenterPointTemplate
MapPolygon.UseCustomPolygonTemplate
MapPolygon.VectorData

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>
```

2.253.1 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.

Parent elements
MapPolygon

The following is the XML Schema definition of the **MapPolygon.MapMarkerTemplate** element.

```
<xsd:element name="MapMarkerTemplate" type="MapMarkerTemplateType"
             minOccurs="0" />
```

2.253.2 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" />
```

2.253.3 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" />
```

2.253.4 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" />
```

2.253.5 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
MapPolygon

The following is the XML Schema definition of the **MapPolygon.UseCustomPolygonTemplate** element.

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

2.253.6 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" />
```

2.254 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
MapPolygonLayer

Child elements
MapPolygonTemplate.ActionInfo
MapPolygonTemplate.CenterPointOffsetX
MapPolygonTemplate.CenterPointOffsetY
MapPolygonTemplate.DataElementLabel
MapPolygonTemplate.DataElementName
MapPolygonTemplate.DataElementOutput
MapPolygonTemplate.Hidden
MapPolygonTemplate.Label
MapPolygonTemplate.LabelPlacement
MapPolygonTemplate.OffsetX
MapPolygonTemplate.OffsetY
MapPolygonTemplate.ScaleFactor
MapPolygonTemplate.ShowLabel
MapPolygonTemplate.Style
MapPolygonTemplate.ToolTip

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>
```

2.254.1 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
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.254.2 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" />
```

2.254.3 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" />
```

2.254.4 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" />
```

2.254.5 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" />
```

2.254.6 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>
```

2.254.7 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.

Parent elements
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.Hidden** element.

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

2.254.8 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.

Parent elements
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.Label** element.

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

2.254.9 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
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.LabelPlacement** element.

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

2.254.10 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
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.OffsetX** element.

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

2.254.11 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
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.OffsetY** element.

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

2.254.12 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
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.ScaleFactor** element.

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

2.254.13 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
MapPolygonTemplate

The following is the XML Schema definition of the **MapPolygonTemplate.ShowLabel** element.

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

2.254.14 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" />
```

2.254.15 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" />
```

2.255 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
MapLayers

Attributes
MapTileLayer.Name

Child elements
MapTileLayer.MapTiles
MapTileLayer.MaximumZoom
MapTileLayer.MinimumZoom
MapTileLayer.TileStyle
MapTileLayer.Transparency
MapTileLayer.UseSecureConnection
MapTileLayer.VisibilityMode

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>
```

2.255.1 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" />
```

2.255.2 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
MapTileLayer

The following is the XML Schema definition of the **MapTileLayer.MapTiles** element.

```
<xsd:element name="MapTiles" type="MapTilesType" minOccurs="0" />
```

2.255.3 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
MapTileLayer

The following is the XML Schema definition of the **MapTileLayer.MaximumZoom** element.

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

2.255.4 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
MapTileLayer

The following is the XML Schema definition of the **MapTileLayer.MinimumZoom** element.

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

2.255.5 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
MapTileLayer

The following is the XML Schema definition of the **MapTileLayer.TileStyle** element.

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

2.255.6 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
MapTileLayer

The following is the XML Schema definition of the **MapTileLayer.Transparency** element.

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

2.255.7 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" />
```

2.255.8 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" />
```

2.256 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

Child elements
MapTiles.MapTile

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>
```

2.256.1 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.

Parent elements
MapTiles

The following is the XML Schema definition of the **MapTiles.MapTile** element.

```
<xsd:element name="MapTile" type="MapTileType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.257 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.

Parent elements
MapTiles

Child elements
MapTile.Name
MapTile.TileData

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>
```

2.257.1 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" />
```

2.257.2 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" />
```

2.258 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
MapLegends.MapLegend

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>
```

2.258.1 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
MapLegends

The following is the XML Schema definition of the **MapLegends.MapLegend** element.

```
<xsd:element name="MapLegend" type="MapLegendType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.259 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
MapLegends

Attributes
MapLegend.Name

Child elements
MapLegend.ActionInfo
MapLegend.AutoFitTextDisabled
MapLegend.BottomMargin
MapLegend.DockOutsideViewport
MapLegend.EquallySpacedItems
MapLegend.Hidden
MapLegend.InterlacedRows
MapLegend.InterlacedRowsColor
MapLegend.Layout
MapLegend.LeftMargin
MapLegend.MapLegendTitle
MapLegend.MapLocation
MapLegend.MapSize
MapLegend.MinFontSize
MapLegend.Position
MapLegend.RightMargin
MapLegend.Style
MapLegend.TextWrapThreshold
MapLegend.ToolTip
MapLegend.TopMargin
MapLegend.ZIndex

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-->
```

```
    <!--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>
```

2.259.1 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 ([XMLSCHEMA2/2] 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" />
```

2.259.2 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" />
```

2.259.3 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" />
```

2.259.4 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" />
```

2.259.5 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" />
```

2.259.6 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" />
```

2.259.7 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" />
```

2.259.8 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" />
```

2.259.9 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" />
```

2.259.10 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" />
```

2.259.11 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" />
```

2.259.12 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" />
```

2.259.13 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" />
```

2.259.14 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" />
```

2.259.15 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" />
```

2.259.16 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" />
```

2.259.17 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" />
```

2.259.18 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" />
```

2.259.19 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" />
```

2.259.20 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" />
```

2.259.21 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
MapLegend

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

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

2.259.22 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
MapLegend

The following is the XML Schema definition of the **MapLegend.ZIndex** element.

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

2.260 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
MapLegend

Child elements
MapLegendTitle.Caption

Child elements
MapLegendTitle.Style
MapLegendTitle.TitleSeparator
MapLegendTitle.TitleSeparatorColor

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>
```

2.260.1 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
MapLegendTitle

The following is the XML Schema definition of the **MapLegendTitle.Caption** element.

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

2.260.2 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" />
```

2.260.3 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" />
```

2.260.4 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" />
```

2.261 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
Map

Child elements
MapTitles.MapTitle

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>
```

2.261.1 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
MapTitles

The following is the XML Schema definition of the **MapTitles.MapTitle** element.

```
<xsd:element name="MapTitle" type="MapTitleType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.262 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
MapTitles

Attributes
MapTitle.Name

Child elements
MapTitle.ActionInfo
MapTitle.Angle
MapTitle.BottomMargin
MapTitle.DockOutsideViewport
MapTitle.Hidden
MapTitle.LeftMargin
MapTitle.MapLocation
MapTitle.MapSize
MapTitle.Position
MapTitle.RightMargin
MapTitle.Style
MapTitle.Text
MapTitle.TextShadowOffset
MapTitle.ToolTip
MapTitle.TopMargin
MapTitle.ZIndex

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: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>
```

2.262.1 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 ([XMLSCHEMA2/2] 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" />
```

2.262.2 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" />
```

2.262.3 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
MapTitle

The following is the XML Schema definition of the **MapTitle.Angle** element.

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

2.262.4 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
MapTitle

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

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

2.262.5 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" />
```

2.262.6 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" />
```

2.262.7 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" />
```

2.262.8 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" />
```

2.262.9 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" />
```

2.262.10 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" />
```

2.262.11 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" />
```

2.262.12 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" />
```

2.262.13 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" />
```

2.262.14 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" />
```

2.262.15 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" />
```

2.262.16 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" />
```

2.262.17 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" />
```

2.263 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
Map

Child elements
MapViewport.BottomMargin
MapViewport.ContentMargin
MapViewport.GridUnderContent
MapViewport.LeftMargin
MapViewport.MapCoordinateSystem
MapViewport.MapLimits
MapViewport.MapLocation
MapViewport.MapMeridians
MapViewport.MapParallels
MapViewport.MapProjection
MapViewport.MapSize
MapViewport.MapCustomView
MapViewport.MapDataBoundView
MapViewport.MapElementView
MapViewport.MaximumZoom
MapViewport.MinimumZoom
MapViewport.ProjectionCenterX
MapViewport.ProjectionCenterY
MapViewport.RightMargin
MapViewport.SimplificationResolution
MapViewport.Style
MapViewport.TopMargin
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>
```

2.263.1 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" />
```

2.263.2 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
MapViewport

The following is the XML Schema definition of the **MapViewport.ContentMargin** element.

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

2.263.3 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
MapViewport

The following is the XML Schema definition of the **MapViewport.GridUnderContent** element.

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

2.263.4 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
MapViewport

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

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

2.263.5 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
MapViewport

The following is the XML Schema definition of the **MapViewport.MapCoordinateSystem** element.

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

2.263.6 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
MapViewport

The following is the XML Schema definition of the **MapViewport.MapLimits** element.

```
<xsd:element name="MapLimits" type="MapLimitsType" minOccurs="0" />
```

2.263.7 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" />
```

2.263.8 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
MapViewport

The following is the XML Schema definition of the **MapViewport.MapMeridians** element.

```
<xsd:element name="MapMeridians" type="MapGridLinesType" minOccurs="0" />
```

2.263.9 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
MapViewport

The following is the XML Schema definition of the **MapViewport.MapParallels** element.

```
<xsd:element name="MapParallels" type="MapGridLinesType" minOccurs="0" />
```

2.263.10 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:

Equirectangular: Equirectangular 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
MapViewport

The following is the XML Schema definition of the **MapViewport.MapProjection** element.

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

2.263.11 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
MapViewport

The following is the XML Schema definition of the **MapViewport.MapSize** element.

```
<xsd:element name="MapSize" type="MapSizeType" minOccurs="0" />
```

2.263.12 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" />
```

2.263.13 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" />
```

2.263.14 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" />
```

2.263.15 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
MapViewport

The following is the XML Schema definition of the **MapViewport.MaximumZoom** element.

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

2.263.16 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
MapViewport

The following is the XML Schema definition of the **MapViewport.MinimumZoom** element.

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

2.263.17 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" />
```

2.263.18 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" />
```

2.263.19 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" />
```

2.263.20 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
MapViewport

The following is the XML Schema definition of the **MapViewport.SimplificationResolution** element.

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

2.263.21 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
MapViewport

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

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

2.263.22 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
MapViewport

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

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

2.263.23 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
MapViewport

The following is the XML Schema definition of the **MapViewport.ZIndex** element.

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

2.264 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
MapViewport

Child elements
MapCustomView.CenterX
MapCustomView.CenterY
MapCustomView.Zoom

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>
```

2.264.1 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
MapCustomView

The following is the XML Schema definition of the **MapCustomView.CenterX** element.

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

2.264.2 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
MapCustomView

The following is the XML Schema definition of the **MapCustomView.CenterY** element.

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

2.264.3 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
MapCustomView

The following is the XML Schema definition of the **MapCustomView.Zoom** element.

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

2.265 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
MapViewport

Child elements
MapCustomView.Zoom

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>
```

2.265.1 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
MapDataBoundView

The following is the XML Schema definition of the **MapDataBoundView.Zoom** element.

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

2.266 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
MapViewport

Child elements
MapElementView.LayerName
MapElementView.MapBindingFieldPairs
MapCustomView.Zoom

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>
```

2.266.1 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
MapElementView

The following is the XML Schema definition of the **MapElementView.LayerName** element.

```
<xsd:element name="LayerName" type="xsd:string" minOccurs="1" />
```

2.266.2 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
MapElementView

The following is the XML Schema definition of the **MapElementView.MapBindingFieldPairs** element.

```
<xsd:element name="MapBindingFieldPairs" type="MapBindingFieldPairsType" minOccurs="0" />
```

2.266.3 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
MapElementView

The following is the XML Schema definition of the **MapElementView.Zoom** element.

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

2.267 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
MapLimits.MaximumX
MapLimits.MaximumY
MapLimits.MinimumX
MapLimits.MinimumY

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>
```

2.267.1 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
MapLimits

The following is the XML Schema definition of the **MapLimits.MaximumX** element.

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

2.267.2 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
MapLimits

The following is the XML Schema definition of the **MapLimits.MaximumY** element.

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

2.267.3 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
MapLimits

The following is the XML Schema definition of the **MapLimits.MinimumX** element.

```
<xsd:element name="MinimumX" type="xsd:string" minOccurs="0" />
```

2.267.4 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
MapLimits

The following is the XML Schema definition of the **MapLimits.MinimumY** element.

```
<xsd:element name="MinimumY" type="xsd:string" minOccurs="0" />
```

2.268 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
MapViewport

Child elements
MapMeridians.Hidden
MapMeridians.Interval
MapMeridians.LabelPosition
MapMeridians.ShowLabels

Child elements
MapMeridians.Style

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>
```

2.268.1 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 ([XMLSCHEMA2/2] 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.Hidden** element.

Parent elements
MapMeridians

The following is the XML Schema definition of the **MapMeridians.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

2.268.2 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.

Parent elements
MapMeridians

The following is the XML Schema definition of the **MapMeridians.Interval** element.

```
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
```

2.268.3 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
MapMeridians

The following is the XML Schema definition of the **MapMeridians.LabelPosition** element.

```
<xsd:element name="LabelPosition" type="xsd:string" minOccurs="0" />
```

2.268.4 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
MapMeridians

The following is the XML Schema definition of the **MapMeridians.ShowLabels** element.

```
<xsd:element name="ShowLabels" type="xsd:string" minOccurs="0" />
```

2.268.5 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
MapMeridians

The following is the XML Schema definition of the **MapMeridians.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

2.269 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
MapViewport

Child elements
MapParallels.Hidden
MapParallels.Interval
MapParallels.LabelPosition
MapParallels.ShowLabels
MapParallels.Style

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>
```

2.269.1 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
MapParallels

The following is the XML Schema definition of the **MapParallels.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0" />
```

2.269.2 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
MapParallels

The following is the XML Schema definition of the **MapParallels.Interval** element.

```
<xsd:element name="Interval" type="xsd:string" minOccurs="0" />
```

2.269.3 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 ([XMLSCHEMA2/2] 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 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.

Parent elements
MapParallels

The following is the XML Schema definition of the **MapParallels.LabelPosition** element.

```
<xsd:element name="LabelPosition" type="xsd:string" minOccurs="0" />
```

2.269.4 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.

Parent elements
MapParallels

The following is the XML Schema definition of the **MapParallels.ShowLabels** element.

```
<xsd:element name="ShowLabels" type="xsd:string" minOccurs="0" />
```

2.269.5 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.

Parent elements
MapParallels

The following is the XML Schema definition of the **MapParallels.Style** element.

```
<xsd:element name="Style" type="StyleType" minOccurs="0" />
```

2.270 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
CustomData.DataColumnHierarchy
CustomData.DataRowHierarchy
CustomData.DataRows
CustomData.DataSetName
CustomData.Filters
CustomData.SortExpressions
CustomData.DataColumnGroupings
CustomData.DataRowGroupings

Applies to RDL 2011/01

Child elements
CustomData.Relationship

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>
```

2.270.1 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
CustomData

The following is the XML Schema definition of the **CustomData.DataColumnHierarchy** element.

```
<xsd:element name="DataColumnHierarchy" type="DataColumnHierarchyType"
             minOccurs="0" />
```

2.270.2 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
CustomData

The following is the XML Schema definition of the **CustomData.DataRowHierarchy** element.

```
<xsd:element name="DataRowHierarchy" type="DataRowHierarchyType" minOccurs="0" />
```

2.270.3 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
CustomData

The following is the XML Schema definition of the **CustomData.DataRows** element.

```
<xsd:element name="DataRows" type="DataRowsType" minOccurs="0" />
```

2.270.4 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
CustomData

The following is the XML Schema definition of the **CustomData.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" />
```

2.270.5 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
CustomData

The following is the XML Schema definition of the **CustomData.Filters** element.

```
<xsd:element name="Filters" type="FiltersType" minOccurs="0" />
```

2.270.6 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
CustomData

The following is the XML Schema definition of the **CustomData.SortExpressions** element.

```
<xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
```

2.270.7 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
CustomData

The following is the XML Schema definition of the **CustomData.DataColumnGroupings** element.

```
<xsd:element name="DataColumnGroupings" type="DataColumnGroupingsType"
             minOccurs="0" />
```

2.270.8 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
CustomData

The following is the XML Schema definition of the **CustomData.DataRowGroupings** element.

```
<xsd:element name="DataRowGroupings" type="DataRowGroupingsType" minOccurs="0" />
```

2.270.9 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" />
```

2.271 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
CustomData

Child element
DataColumnHierarchy.DataMembers

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>
```

2.271.1 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
DataColumnHierarchy

The following is the XML Schema definition of the **DataColumnHierarchy.DataMembers** element.

```
<xsd:element name="DataMembers" type="DataMembersType"/>
```

2.272 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
DataColumnHierarchy
DataRowHierarchy

Child element
DataMembers.DataMember

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>
```

2.272.1 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" />
```

2.273 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
DataMembers

Child elements
DataMember.CustomProperties
DataMember.DataMembers
DataMember.Group
DataMember.SortExpressions
DataMember.Subtotal

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>
```

2.273.1 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" />
```

2.273.2 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" />
```

2.273.3 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" />
```

2.273.4 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
DataMember

The following is the XML Schema definition of the **DataMember.SortExpressions** element.

```
<xsd:element name="SortExpressions" type="SortExpressionsType" minOccurs="0" />
```

2.273.5 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
DataMember

The following is the XML Schema definition of the **DataMember.Subtotal** element.

```
<xsd:element name="Subtotal" type="xsd:boolean" minOccurs="0" />
```

2.274 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
CustomData

Child element
DataRowHierarchy.DataMembers

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>
```

2.274.1 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.

Parent elements
DataRowHierarchy

The following is the XML Schema definition of the **DataRowHierarchy.DataMembers** element.

```
<xsd:element name="DataMembers" type="DataMembersType"/>
```

2.275 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.

Parent elements
CustomData

Child element
DataRows.DataRow

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>
```

2.275.1 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.

Parent elements
DataRows

The following is the XML Schema definition of the **DataRows.DataRow** element.

```
<xsd:element name="DataRow" type="DataRowType" maxOccurs="unbounded" />
```

2.276 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.

Parent elements
DataRows

Child element
DataRow.DataCell

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>
```

2.276.1 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.

Parent elements
DataRow

The following is the XML Schema definition of the **DataRow.DataCell** element.

```
<xsd:element name="DataCell" type="DataCellType" maxOccurs="unbounded" />
```

2.277 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
DataRow

Child elements
DataCell.DataValue

Applies to RDL 2011/01

Child elements
DataCell.DataSetName
DataCell.Relationships

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>
```

2.277.1 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.

Parent elements
DataCell

The following is the XML Schema definition of the **DataCell.DataValue** element.

```
<xsd:element name="DataValue" type="DataValueType" maxOccurs="unbounded" />
```

2.277.2 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.

Parent elements
DataCell

The following is the XML Schema definition of the **DataCell.DataSetName** element.

```
<xsd:element name ="DataSetName" type="xsd:string" minOccurs="0"/>
```

2.277.3 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.

Parent elements
DataCell

The following is the XML Schema definition of the **DataCell.Relationships** element.

```
<xsd:element name ="Relationships" type="RelationshipsType" minOccurs="0" />
```

2.278 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
DataCell
DataValues

Child elements
DataValue.Name
DataValue.Value

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>
```

2.278.1 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
DataValue

The following is the XML Schema definition of the **DataValue.Name** element.

```
<xsd:element name="Name" type="xsd:string" minOccurs="0" />
```

2.278.2 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
DataValue

The following is the XML Schema definition of the **DataValue.Value** element.

```
<xsd:element name="Value" type="xsd:string" />
```

2.279 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
CustomData

Child element
DataColumnGroupings.DataGroupings

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>
```

2.279.1 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
DataColumnGroupings

The following is the XML Schema definition of the **DataColumnGroupings.DataGroupings** element.

```
<xsd:element name="DataGroupings" type="DataGroupingsType" />
```

2.280 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
DataColumnGroupings
DataRowGroupings

Child element
DataGroupings.DataGrouping

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>
```

2.280.1 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
DataGroupings

The following is the XML Schema definition of the **DataGroupings.DataGrouping** element.

```
<xsd:element name="DataGrouping" type="DataGroupingType" maxOccurs="unbounded" />
```

2.281 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.

Parent elements
DataGroupings

Child elements
DataGrouping.CustomProperties
DataGrouping.DataGroupings
DataGrouping.Grouping
DataGrouping.Sorting
DataGrouping.Static
DataGrouping.Subtotal

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>
```

2.281.1 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.

Parent elements
DataGrouping

The following is the XML Schema definition of the **DataGrouping.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

2.281.2 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" />
```

2.281.3 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" />
```

2.281.4 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" />
```

2.281.5 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
DataGrouping

The following is the XML Schema definition of the **DataGrouping.Static** element.

```
<xsd:element name="Static" type="xsd:boolean" minOccurs="0" />
```

2.281.6 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
DataGrouping

The following is the XML Schema definition of the **DataGrouping.Subtotal** element.

```
<xsd:element name="Subtotal" type="xsd:boolean" minOccurs="0" />
```

2.282 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
CustomData

Child element
DataRowGroupings.DataGroupings

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>
```

2.282.1 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
DataRowGroupings

The following is the XML Schema definition of the **DataRowGroupings.DataGroupings** element.

```
<xsd:element name="DataGroupings" type="DataGroupingsType" />
```

2.283 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
ReportItems

The following are additional parent elements of the **CustomReportItem** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Parent elements
CellContents

The following are the attributes of the **CustomReportItem** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Attributes
CustomReportItem.Name

The following are the child elements of the **CustomReportItem** element.

Child elements
CustomReportItem.Type

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
CustomReportItem.Label

The following are additional child elements of the **CustomReportItem** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Child elements
CustomReportItem.ActionInfo

Child elements
CustomReportItem.AltReportItem
CustomReportItem.Bookmark
CustomReportItem.CustomData
CustomReportItem.CustomProperties
CustomReportItem.DataElementName
CustomReportItem.DataElementOutput
CustomReportItem.DocumentMapLabel
CustomReportItem.Height
CustomReportItem.Left
CustomReportItem.RepeatWith
CustomReportItem.Style
CustomReportItem.ToolTip
CustomReportItem.Top
CustomReportItem.Type
CustomReportItem.Visibility
CustomReportItem.Width
CustomReportItem.ZIndex

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: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 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: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>
```

2.283.1 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" />
```

2.283.2 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
CustomReportItem

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" />
```

2.283.3 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
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.ActionInfo** element.

```
<xsd:element name="ActionInfo" type="ActionInfoType" minOccurs="0" />
```

2.283.4 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" />
```

2.283.5 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" />
```

2.283.6 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" />
```

2.283.7 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
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.CustomProperties** element.

```
<xsd:element name="CustomProperties" type="CustomPropertiesType" minOccurs="0" />
```

2.283.8 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
CustomReportItem

The following is the XML Schema definition of the **CustomReportItem.DataElementName** element.

```
<xsd:element name="DataElementName" type="xsd:string" minOccurs="0" />
```

2.283.9 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
CustomReportItem

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>
```

2.283.10 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" />
```

2.283.11 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" />
```

2.283.12 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" />
```

2.283.13 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" />
```

2.283.14 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" />
```

2.283.15 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" />
```

2.283.16 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" />
```

2.283.17 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" />
```

2.283.18 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" />
```

2.283.19 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" />
```

2.283.20 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" />
```

2.283.21 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" />
```

2.284 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
Image

Parent elements
Line
Rectangle
Subreport
Textbox
TextRun
Tablix
Chart
ChartStripLine
ChartDataLabel
ChartDataPoint
ChartEmptyPoints
ChartItemInLegend
ChartLegendColumn
ChartTitle
ChartLegendCustomItem
ChartLegendCustomItemCell
GaugePanel
GaugeImage
GaugeLabel
LinearGauge
LinearScale
LinearPointer
ScaleRange
NumericIndicator
RadialGauge
RadialScale
RadialPointer
StateIndicator
Map
MapColorScale
MapDistanceScale
MapLineTemplate

Parent elements
MapMarkerTemplate
MapPolygonTemplate
MapLegend
MapTitle
CustomReportItem

Child elements
ActionInfo.Actions

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>
```

2.284.1 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
ActionInfo

The following is the XML Schema definition of the **ActionInfo.Actions** element.

```
<xsd:element name="Actions" type="ActionsType" minOccurs="0">
```

2.285 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
ActionInfo

Child elements
Actions.Action

The following is the XML Schema definition of the **Actions** element in RDL 2008/01.

```
<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>
```

The following is the XML Schema definition of the **Actions** element in RDL 2010/01 and RDL 2016/01.

```
<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>
```

2.285.1 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" />
```

2.286 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
Image
Line
Rectangle
Subreport
Textbox
Chart

The following are additional parent elements of the **Action** element in RDL 2003/10 and RDL 2005/01.

Parent elements
List
Matrix
Table
DataPoint

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>
```

2.286.1 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
Action

The following is the XML Schema definition of the **Action.BookmarkLink** element.

```
<xsd:element name="BookmarkLink" type="xsd:string" minOccurs="0">
```

2.286.2 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
Action

The following is the XML Schema definition of the **Action.Drillthrough** element.

```
<xsd:element name="Drillthrough" type="DrillthroughType" minOccurs="0">
```

2.286.3 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
Action

The following is the XML Schema definition of the **Action.Hyperlink** element.

```
<xsd:element name="Hyperlink" type="xsd:string" minOccurs="0">
```

2.286.4 Action.Label

Applies to RDL 2005/01

The **Action.Label** element is ignored.

Following is the parent element of the **Action.Label** element.

Parent elements
Action

The following is the XML Schema definition of the **Action.Label** element.

```
<xsd:element name="Label" type="xsd:string" minOccurs="0">
```

2.287 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.

Parent elements
Action

Child elements
Drillthrough.Parameters
Drillthrough.ReportName

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>
```

2.287.1 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.

Parent elements
Drillthrough

The following is the XML Schema definition of the **Drillthrough.Parameters** element.

```
<xsd:element name="Parameters" type="ParametersType" minOccurs="0">
```

2.287.2 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
Drillthrough

The following is the XML Schema definition of the **Drillthrough.ReportName** element.

```
<xsd:element name="ReportName" type="xsd:string">
```

2.288 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
Report
Image
Line
Rectangle
Subreport
Textbox
List
Matrix
Table
Grouping
Chart

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>
```

2.289 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
Report
Image
Line
Rectangle
Subreport
Textbox
Tablix
TablixMember
List
Matrix
Table
Grouping
Chart
ChartAxis
ChartSeries
ChartDataPoint
ChartEmptyPoints
ChartMember
GaugePanel
Map
CustomReportItem
DataMember
DataGrouping

Child elements
CustomProperties.CustomProperty

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>
```

2.289.1 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.

Parent elements
CustomProperties

The following is the XML Schema definition of the **CustomProperties.CustomProperty** element.

```
<xsd:element name="CustomProperty" type="CustomPropertyType" maxOccurs="unbounded">
```

2.290 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.

Parent elements
CustomProperties

Child elements
CustomProperty.Name
CustomProperty.Value

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>
```

2.290.1 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.

Parent elements
CustomProperty

The following is the XML Schema definition of the **CustomProperty.Name** element.

```
<xsd:element name="Name" type="xsd:string">
```

2.290.2 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
CustomProperty

The following is the XML Schema definition of the **CustomProperty.Value** element.

```
<xsd:element name="Value" type="xsd:string">
```

2.291 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
Rectangle
Group
Chart
Tablix
GaugePanel
Map

Child elements
PageBreak.BreakLocation
PageBreak.Disabled
PageBreak.ResetPageNumber

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: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>
```

2.291.1 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.

Parent elements
PageBreak

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>
```

2.291.2 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
PageBreak

The following is the XML Schema definition of the **PageBreak.Disabled** element.

```
<xsd:element name="Disabled" type="xsd:string" minOccurs="0" />
```

2.291.3 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
PageBreak

The following is the XML Schema definition of the **PageBreak.ResetPageNumber** element.

```
<xsd:element name="ResetPageNumber" type="xsd:string" minOccurs="0" />
```

2.292 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
Line
Rectangle
Textbox
Paragraph
TextRun
Image
SubReport
Tablix
Chart
Body
PageSection
Page
ChartAxis
ChartBorderSkin
ChartTitle
ChartLegend
ChartLegendTitle
ChartArea
ChartAxisTitle
ChartAxisScaleBreak
ChartSeries
ChartDataPoint
ChartEmptyPoints
ChartDataLabel
ChartMarker
ChartGridLines
ChartTickMarks
ChartStripLine
LinearGauge
RadialGauge
GaugePanel

Parent elements
LinearPointer
RadialPointer
ScaleRange
LinearScale
RadialScale
Thermometer
PointerCap
ScaleLabels
CustomLabel
TickMarkStyle
GaugeTickMarks
ScalePin
PinLabel
BackFrame
FrameBackground
ChartLegendColumn
ChartLegendColumnHeader
ChartLegendCustomItem
ChartLegendCustomItemCell
Axis
DataLabel
DataPoint
MajorGridLines
Marker
MinorGridLines
PlotArea
Title
GaugeLabel
NumericIndicator
StateIndicator
IndicatorState
CustomReportItem

Parent elements
List
Matrix
Table
Subtotal
Map
MapBorderSkin
MapColorScale
MapColorScaleTitle
MapDistanceScale
MapLegend
MapLegendTitle
MapLineTemplate
MapMarkerTemplate
MapPolygonTemplate
MapTitle
MapViewport

Child elements
Style.BackgroundColor
Style.BackgroundGradientEndColor
Style.BackgroundGradientType
Style.BackgroundHatchType
Style.BackgroundImage
Style.Border
Style.BottomBorder
Style.Calendar
Style.Color
Style.Direction
Style.FontFamily
Style.FontSize
Style.FontStyle

Child elements
Style.FontWeight
Style.Format
Style.Language
Style.LeftBorder
Style.LineHeight
Style.NumeralLanguage
Style.NumeralVariant
Style.PaddingBottom
Style.PaddingLeft
Style.PaddingRight
Style.PaddingTop
Style.RightBorder
Style.ShadowColor
Style.ShadowOffset
Style.TextAlign
Style.TextDecoration
Style.TextEffect
Style.TopBorder
Style.VerticalAlign
Style.WritingMode
Style.UnicodeBiDi
Style.BorderColor
Style.BorderStyle
Style.BorderWidth

Applies to RDL 2013/01

Child elements
Style.CurrencyLanguage

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: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: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>
```

2.292.1 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 **RdlColor**.

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
Style

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" />
```

2.292.2 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 RdlColor value or an expression that evaluates to an **RdlColor**.

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
Style

The following is the XML Schema definition of the **Style.BackgroundGradientEndColor** element.

```
<xsd:element name="BackgroundGradientEndColor" type="xsd:string" minOccurs="0" />
```

2.292.3 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
Style

The following is the XML Schema definition of the **Style.BackgroundGradientType** element.

```
<xsd:element name="BackgroundGradientType" type="xsd:string" minOccurs="0" />
```

2.292.4 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
Style

The following is the XML Schema definition of the **Style.BackgroundHatchType** element.

```
<xsd:element name="BackgroundHatchType" type="xsd:string" minOccurs="0" />
```

2.292.5 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
Style

The following is the XML Schema definition of the **Style.BackgroundImage** element.

```
<xsd:element name="BackgroundImage" type="BackgroundImageType" minOccurs="0" />
```

2.292.6 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
Style

The following is the XML Schema definition of the **Style.Border** element.

```
<xsd:element name="Border" type="BorderType" minOccurs="0" />
```

2.292.7 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
Style

The following is the XML Schema definition of the **Style.BorderColor** element.

```
<xsd:element name="BorderColor" type="BorderColorStyleWidthType" minOccurs="0" />
```

2.292.8 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
Style

The following is the XML Schema definition of the **Style.BorderStyle** element.

```
<xsd:element name="BorderStyle" type="BorderColorStyleWidthType" minOccurs="0" />
```

2.292.9 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
Style

The following is the XML Schema definition of the **Style.BorderWidth** element.

```
<xsd:element name="BorderWidth" type="BorderColorStyleWidthType" minOccurs="0" />
```

2.292.10 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
Style

The following is the XML Schema definition of the **Style.BottomBorder** element.

```
<xsd:element name="BottomBorder" type="BorderType" minOccurs="0" />
```

2.292.11 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
Style

The following is the XML Schema definition of the **Style.Calendar** element.

```
<xsd:element name="Calendar" type="xsd:string" minOccurs="0" />
```

2.292.12 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
Style

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" />
```

2.292.13 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
Style

The following is the XML Schema definition of the **Style.CurrencyLanguage** element.

```
<xsd:element name="CurrencyLanguage" type="xsd:string" />
```

2.292.14 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
Style

The following is the XML Schema definition of the **Style.Direction** element.

```
<xsd:element name="Direction" type="xsd:string" minOccurs="0" />
```

2.292.15 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
Style

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" />
```

2.292.16 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
Style

The following is the XML Schema definition of the **Style.FontSize** element.

```
<xsd:element name="FontSize" type="xsd:string" minOccurs="0" />
```

2.292.17 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
Style

The following is the XML Schema definition of the **Style.FontStyle** element.

```
<xsd:element name="FontStyle" type="xsd:string" minOccurs="0" />
```

2.292.18 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
Style

The following is the XML Schema definition of the **Style.FontWeight** element.

```
<xsd:element name="FontWeight" type="xsd:string" minOccurs="0" />
```

2.292.19 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.

Parent elements
Style

The following is the XML Schema definition of the **Style.Format** element.

```
<xsd:element name="Format" type="xsd:string" minOccurs="0" />
```

2.292.20 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.

Parent elements
Style

The following is the XML Schema definition of the **Style.Language** element.

```
<xsd:element name="Language" type="xsd:string" minOccurs="0" />
```

2.292.21 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.

Parent elements
Style

The following is the XML Schema definition of the **Style.LeftBorder** element.

```
<xsd:element name="LeftBorder" type="BorderType" minOccurs="0" />
```

2.292.22 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, Textboxor Subtotal, the **Style.LineHeight** element is ignored.

Following is the parent element of the **Style.LineHeight** element.

Parent elements
Style

The following is the XML Schema definition of the **Style.LineHeight** element.

```
<xsd:element name="LineHeight" type="xsd:string" minOccurs="0" />
```

2.292.23 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
Style

The following is the XML Schema definition of the **Style.NumeralLanguage** element.

```
<xsd:element name="NumeralLanguage" type="xsd:string" minOccurs="0" />
```

2.292.24 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
Style

The following is the XML Schema definition of the **Style.NumeralVariant** element.

```
<xsd:element name="NumeralVariant" type="xsd:string" minOccurs="0" />
```

2.292.25 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
Style

The following is the XML Schema definition of the **Style.PaddingBottom** element.

```
<xsd:element name="PaddingBottom" type="xsd:string" minOccurs="0" />
```

2.292.26 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
Style

The following is the XML Schema definition of the **Style.PaddingLeft** element.

```
<xsd:element name="PaddingLeft" type="xsd:string" minOccurs="0" />
```

2.292.27 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
Style

The following is the XML Schema definition of the **Style.PaddingRight** element.

```
<xsd:element name="PaddingRight" type="xsd:string" minOccurs="0" />
```

2.292.28 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
Style

The following is the XML Schema definition of the **Style.PaddingTop** element.

```
<xsd:element name="PaddingTop" type="xsd:string" minOccurs="0" />
```

2.292.29 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
Style

The following is the XML Schema definition of the **Style.RightBorder** element.

```
<xsd:element name="RightBorder" type="BorderType" minOccurs="0" />
```

2.292.30 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.

Parent elements
Style

The following is the XML Schema definition of the **Style.ShadowColor** element.

```
<xsd:element name="ShadowColor" type="xsd:string" minOccurs="0" />
```

2.292.31 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.

Parent elements
Style

The following is the XML Schema definition of the **Style.ShadowOffset** element.

```
<xsd:element name="ShadowOffset" type="xsd:string" minOccurs="0" />
```

2.292.32 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.

Parent elements
Style

The following is the XML Schema definition of the **Style.TextAlign** element.

```
<xsd:element name="TextAlign" type="xsd:string" minOccurs="0" />
```

2.292.33 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.

Parent elements
Style

The following is the XML Schema definition of the **Style.TextDecoration** element.

```
<xsd:element name="TextDecoration" type="xsd:string" minOccurs="0" />
```

2.292.34 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
Style

The following is the XML Schema definition of the **Style.TextEffect** element.

```
<xsd:element name="TextEffect" type="xsd:string" minOccurs="0" />
```

2.292.35 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
Style

The following is the XML Schema definition of the **Style.TopBorder** element.

```
<xsd:element name="TopBorder" type="BorderType" minOccurs="0" />
```

2.292.36 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
Style

The following is the XML Schema definition of the **Style.UnicodeBiDi** element.

```
<xsd:element name="UnicodeBiDi" type="xsd:string" minOccurs="0" />
```

2.292.37 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
Style

The following is the XML Schema definition of the **Style.VerticalAlign** element.

```
<xsd:element name="VerticalAlign" type="xsd:string" minOccurs="0" />
```

2.292.38 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
Style

The following is the XML Schema definition of the **Style.WritingMode** element.

```
<xsd:element name="WritingMode" type="xsd:string" minOccurs="0" />
```

2.293 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
Style

Child elements
BackgroundImage.BackgroundRepeat
BackgroundImage.MIMEType
BackgroundImage.Position
BackgroundImage.Source
BackgroundImage.TransparentColor
BackgroundImage.Value

Applies to RDL 2012/01

Child elements
BackgroundImage.EmbeddingMode
BackgroundImage.Transparency

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>
```

2.293.1 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" />
```

2.293.2 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" />
```

2.293.3 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" />
```

2.293.4 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.

Parent elements
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>
```

2.293.5 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.

Parent elements
BackgroundImage

The following is the XML Schema definition of the **BackgroundImage.TransparentColor** element.

```
<xsd:element name="TransparentColor" type="xsd:string" minOccurs="0" />
```

2.293.6 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
BackgroundImage

The following is the XML Schema definition of the **BackgroundImage.Value** element.

```
<xsd:element name="Value" type="xsd:string" />
```

2.293.7 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
BackgroundImage

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>
```

2.293.8 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.

Parent elements
BackgroundImage

The following is the XML Schema definition of the **BackgroundImage.Transparency** element.

```
<xsd:element name="Transparency" type="xsd:string" />
```

2.294 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.

Parent elements
Style

Child elements
Border.Color
Border.Style
Border.Width

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>
```

2.294.1 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" />
```

2.294.2 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" />
```

2.294.3 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
Border

The following is the XML Schema definition of the **Border.Width** element.

```
<xsd:element name="Width" type="xsd:string" minOccurs="0" />
```

2.295 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
Style

Child elements
BorderColor.Bottom
BorderColor.Default
BorderColor.Left
BorderColor.Right
BorderColor.Top

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: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>
```

2.295.1 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
BorderColor

The following is the XML Schema definition of the **BorderColor.Bottom** element.

```
<xsd:element name="Bottom" type="xsd:string" minOccurs="0" />
```

2.295.2 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
BorderColor

The following is the XML Schema definition of the **BorderColor.Default** element.

```
<xsd:element name="Default" type="xsd:string" minOccurs="0" />
```

2.295.3 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.

Parent elements
BorderColor

The following is the XML Schema definition of the **BorderColor.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

2.295.4 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.

Parent elements
BorderColor

The following is the XML Schema definition of the **BorderColor.Right** element.

```
<xsd:element name="Right" type="xsd:string" minOccurs="0" />
```

2.295.5 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.

Parent elements
BorderColor

The following is the XML Schema definition of the **BorderColor.Top** element.

```
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
```

2.296 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.

Parent elements
Style

Child elements
BorderStyle.Bottom
BorderStyle.Default
BorderStyle.Left
BorderStyle.Right
BorderStyle.Top

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="#
```

2.296.1 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.

Parent elements
BorderStyle

The following is the XML Schema definition of the **BorderStyle.Bottom** element.

```
<xsd:element name="Bottom" type="xsd:string" minOccurs="0" />
```

2.296.2 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.

Parent elements
BorderStyle

The following is the XML Schema definition of the **BorderStyle.Default** element.

```
<xsd:element name="Default" type="xsd:string" minOccurs="0" />
```

2.296.3 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
BorderStyle

The following is the XML Schema definition of the **BorderStyle.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

2.296.4 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
BorderStyle

The following is the XML Schema definition of the **BorderStyle.Right** element.

```
<xsd:element name="Right" type="xsd:string" minOccurs="0" />
```

2.296.5 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
BorderStyle

The following is the XML Schema definition of the **BorderStyle.Top** element.

```
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
```

2.297 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
Style

Child elements
BorderWidth.Bottom
BorderWidth.Default
BorderWidth.Left
BorderWidth.Right
BorderWidth.Top

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>
```

2.297.1 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
BorderWidth

The following is the XML Schema definition of the **BorderWidth.Bottom** element.

```
<xsd:element name="Bottom" type="xsd:string" minOccurs="0" />
```

2.297.2 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
BorderWidth

The following is the XML Schema definition of the **BorderWidth.Default** element.

```
<xsd:element name="Default" type="xsd:string" minOccurs="0" />
```

2.297.3 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
BorderWidth

The following is the XML Schema definition of the **BorderWidth.Left** element.

```
<xsd:element name="Left" type="xsd:string" minOccurs="0" />
```

2.297.4 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.

Parent elements
BorderWidth

The following is the XML Schema definition of the **BorderWidth.Right** element.

```
<xsd:element name="Right" type="xsd:string" minOccurs="0" />
```

2.297.5 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 RdlSize or expression that evaluates to an **RdlSize**. 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.

Parent elements
BorderWidth

The following is the XML Schema definition of the **BorderWidth.Top** element.

```
<xsd:element name="Top" type="xsd:string" minOccurs="0" />
```

2.298 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.

Parent elements
Subreport
Drillthrough

Child elements
Parameters.Parameter

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>
```

2.298.1 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
Parameters

The following is the XML Schema definition of the **Parameters.Parameter** element.

```
<xsd:element name="Parameter" type="ParameterType" maxOccurs="unbounded" />
```

2.299 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
Parameter.Name

Child elements
Parameter.Omit
Parameter.Value

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>
```

2.299.1 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.

Parent elements
Parameter

The following is the XML Schema definition of the **Parameter.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:string" use="required" />
```

2.299.2 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.

Parent elements
Parameter

The following is the XML Schema definition of the **Parameter.Omit** element.

```
<xsd:element name="Omit" type="xsd:string" minOccurs="0" />
```

2.299.3 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.

Parent elements
Parameter

The following is the XML Schema definition of the **Parameter.Value** element.

```
<xsd:element name="Value" type="xsd:string" />
```

2.300 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
Image
Line
Rectangle
Subreport
Textbox
Tablix
TablixMember
List
Matrix
DynamicColumns
DynamicRows
Table
Details
TableRow
TableColumn
TableGroup
Chart
GaugePanel
Map
CustomReportItem

Child elements
Visibility.Hidden
Visibility.ToggleItem

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>
```

2.300.1 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
Visibility

The following is the XML Schema definition of the **Visibility.Hidden** element.

```
<xsd:element name="Hidden" type="xsd:string" minOccurs="0">
```

2.300.2 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
Visibility

The following is the XML Schema definition of the **Visibility.ToggleItem** element.

```
<xsd:element name="ToggleItem" type="xsd:string" minOccurs="0">
```

2.301 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
Classes.Class

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>
```

2.301.1 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
Classes

The following is the XML Schema definition of the **Classes.Class** element.

```
<xsd:element name="Class" type="ClassType" maxOccurs="unbounded">
```

2.302 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
Classes

Child elements
Class.ClassName
Class.InstanceName

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:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
 </xsd:complexType>
```

2.302.1 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
Class

The following is the XML Schema definition of the **Class.ClassName** element.

```
<xsd:element name="ClassName" type="xsd:string">
```

2.302.2 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
Class

The following is the XML Schema definition of the **Class.InstanceName** element.

```
<xsd:element name="InstanceName" type="xsd:normalizedString">
```

2.303 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

Child elements
CodeModules.CodeModule

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>
```

2.303.1 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.

Parent elements
CodeModules

The following is the XML Schema definition of the **CodeModules.CodeModule** element.

```
<xsd:element name="CodeModule" type="xsd:string" maxOccurs="unbounded">
```

2.304 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.

Parent elements
Report

Child elements
EmbeddedImages.EmbeddedImage

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>
```

2.304.1 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
EmbeddedImages

The following is the XML Schema definition of the **EmbeddedImages.EmbeddedImage** element.

```
<xsd:element name="EmbeddedImage" type="EmbeddedImageType" maxOccurs="unbounded">
```

2.305 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
EmbeddedImages

Attributes
EmbeddedImage.Name

Child elements
EmbeddedImage.ImageData
EmbeddedImage.MIMEType

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>
```

2.305.1 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" />
```

2.305.2 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">
```

2.305.3 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">
```

2.306 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
ReportParameters.ReportParameter

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>
```

2.306.1 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" />
```

2.307 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
ReportParameter.Name

Child elements
ReportParameter.AllowBlank
ReportParameter.DataType
ReportParameter.DefaultValue
ReportParameter.Hidden
ReportParameter.MultiValue
ReportParameter.Nullable
ReportParameter.Prompt
ReportParameter.UsedInQuery
ReportParameter.ValidValues

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>
```

2.307.1 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" />
```

2.307.2 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" />
```

2.307.3 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>
```

2.307.4 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" />
```

2.307.5 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" />
```

2.307.6 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" />
```

2.307.7 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" />
```

2.307.8 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" />
```

2.307.9 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>
```

2.307.10 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" />
```

2.308 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
ReportParametersLayout.GridLayoutDefinition

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>
```

2.308.1 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.

Parent elements
ReportParametersLayout

The following is the XML Schema definition of the **ReportParametersLayout.GridLayoutDefinition** element.

```
<xsd:element name="GridLayoutDefinition" type="GridLayoutDefinitionType" minOccurs="1" />
```

2.309 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.

Parent elements
ReportParametersLayout

Child elements
GridLayoutDefinition.NumberOfColumns
GridLayoutDefinition.NumberOfRows
GridLayoutDefinition.CellDefinitions

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>
```

2.309.1 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.

Parent elements
GridLayoutDefinition

The following is the XML Schema definition of the **GridLayoutDefinition.NumberOfColumns** element.

```
<xsd:element name="NumberOfColumns" type="xsd:int" minOccurs="1" />
```

2.309.2 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
GridLayoutDefinition

The following is the XML Schema definition of the **GridLayoutDefinition.NumberOfRows** element.

```
<xsd:element name="NumberOfRows" type="xsd:int" minOccurs="1" />
```

2.309.3 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
GridLayoutDefinition

The following is the XML Schema definition of the **GridLayoutDefinition.CellDefinitions** element.

```
<xsd:element name="CellDefinitions" type="CellDefinitionsType" minOccurs="0" />
```

2.310 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
GridLayoutDefinition

Child elements
CellDefinitions.CellDefinition

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>
```

2.310.1 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.

Parent elements
CellDefinitions

The following is the XML Schema definition of the **CellDefinitions.CellDefinition** element.

```
<xsd:element name="CellDefinition" type="CellDefinitionType" minOccurs="1"
maxOccurs="unbounded" />
```

2.311 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.

Parent elements
CellDefinitions

Child elements
CellDefinition.ColumnIndex
CellDefinition.RowIndex
CellDefinition.ParameterName

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>
```

2.311.1 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.

Parent elements
CellDefinition

The following is the XML Schema definition of the **CellDefinition.ColumnIndex** element.

```
<xsd:element name="ColumnIndex" type="xsd:int" minOccurs="1" />
```

2.311.2 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.

Parent elements
CellDefinition

The following is the XML Schema definition of the **CellDefinition.RowIndex** element.

```
<xsd:element name="RowIndex" type="xsd:int" minOccurs="1" />
```

2.311.3 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.

Parent elements
CellDefinition

The following is the XML Schema definition of the **CellDefinition.ParameterName** element.

```
<xsd:element name="ParameterName" type="xsd:normalizedString" minOccurs="1" />
```

2.312 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.

Parent elements
ReportParameter

Child elements
DefaultValue.DataSetReference
DefaultValue.Values

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:any namespace="##other" processContents="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

2.312.1 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
DefaultValue

The following is the XML Schema definition of the **DefaultValue.DataSetReference** element.

```
<xsd:element name="DataSetReference" type="DataSetReferenceType" minOccurs="0" />
```

2.312.2 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
DefaultValue

The following is the XML Schema definition of the **DefaultValue.Values** element.

```
<xsd:element name="Values" type="ValuesType" minOccurs="0"/>
```

2.313 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
DataSetReference.DataSetName
DataSetReference.LabelField
DataSetReference.ValueField

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>
```

2.313.1 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
DataSetReference

The following is the XML Schema definition of the **DataSetReference.DataSetName** element.

```
<xsd:element name="DataSetName" type="xsd:string" />
```

2.313.2 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
DataSetReference

The following is the XML Schema definition of the **DataSetReference.LabelField** element.

```
<xsd:element name="LabelField" type="xsd:string" minOccurs="0" />
```

2.313.3 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
DataSetReference

The following is the XML Schema definition of the **DataSetReference.ValueField** element.

```
<xsd:element name="ValueField" type="xsd:string" />
```

2.314 Values

The **Values** element specifies a set of values. For a non-multivalue 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
DefaultValue

Child elements
Values.Value

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>
```

2.314.1 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"/>
```

2.315 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>
```

2.315.1 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
ValidValues

The following is the XML Schema definition of the **ValidValues.DataSetReference** element.

```
<xsd:element name="DataSetReference" type="DataSetReferenceType" minOccurs="0" />
```

2.315.2 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
ValidValues

The following is the XML Schema definition of the **ValidValues.ParameterValues** element.

```
<xsd:element name="ParameterValues" type="ParameterValuesType" minOccurs="0" />
```

2.316 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.

Parent elements
ValidValues

Child elements
ParameterValues.ParameterValue

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>
```

2.316.1 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.

Parent elements
ParameterValues

The following is the XML Schema definition of the **ParameterValues.ParameterValue** element.

```
<xsd:element name="ParameterValue" type="ParameterValueType" maxOccurs="unbounded">
```

2.317 ParameterValue

The **ParameterValue** element specifies a value/label pair for ValidValues.

The following are the parent and child elements of the **ParameterValue** element.

Parent elements
ParameterValues

Child elements
ParameterValue.Label
ParameterValue.Value

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>
```

2.317.1 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
ParameterValue

The following is the XML Schema definition of the **ParameterValue.Label** element.

```
<xsd:element name="Label" type=" StringLocIDType" minOccurs="0" />
```

2.317.2 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
ParameterValue

The following is the XML Schema definition of the **ParameterValue.Value** element.

```
<xsd:element name="Value" type="xsd:string" minOccurs="0"/>
```

2.318 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
Group
Report

Child elements
Variables.Variable

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>
```

2.318.1 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
Variables

The following is the XML Schema definition of the **Variables.Variable** element.

```
<xsd:element name="Variable" type="VariableType" minOccurs="1" maxOccurs="unbounded" />
```

2.319 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
Variables

Attributes
Variable.Name

The following are the child elements of the **Variable** element in RDL 2008/01, RDL 2010/01, and RDL 2016/01.

Child elements
Variable.Value
Variable.Value.Datatype

The following are the child elements of the **Variable** element in RDL 2010/01 and RDL 2016/01.

Child elements
Variable.Writable

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: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>
```

2.319.1 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
Variable

The following is the XML Schema definition of the **Variable.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:normalizedString" use="required" />
```

2.319.2 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
Variable

Attributes
Variable.Value.DataType

The following is the XML Schema definition of the **Variable.Value** element.

```
<xsd:element name="Value" type="StringWithDataTypeAttribute"
             minOccurs="1" maxOccurs="1">
```

2.319.3 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] 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 **Variable.Value.DataType** attribute.

Parent elements
Variable.Value

2.319.4 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" />
```

2.320 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
DataSources.DataSource

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>
```

2.320.1 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
DataSources

The following is the XML Schema definition of the **DataSources.DataSource** element.

```
<xsd:element name="DataSource" type="DataSourceType" maxOccurs="unbounded" />
```

2.321 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
DataSources

Attributes
DataSource.Name

Child elements
DataSource.ConnectionProperties
DataSource.DataSourceReference
DataSource.Transaction

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>
```

2.321.1 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
DataSource

The following is the XML Schema definition of the **DataSource.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:string" use="required" />
```

2.321.2 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
DataSource

The following is the XML Schema definition of the **DataSource.ConnectionProperties** element.

```
<xsd:element name="ConnectionProperties" type="ConnectionPropertiesType"
             minOccurs="0" />
```

2.321.3 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
DataSource

The following is the XML Schema definition of the **DataSource.DataSourceReference** element.

```
<xsd:element name="DataSourceReference" type="xsd:string" minOccurs="0" />
```

2.321.4 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
DataSource

The following is the XML Schema definition of the **DataSource.Transaction** element.

```
<xsd:element name="Transaction" type="xsd:boolean" minOccurs="0" />
```

2.322 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
DataSource

Child elements
ConnectionProperties.ConnectString
ConnectionProperties.DataProvider
ConnectionProperties.IntegratedSecurity
ConnectionProperties.Prompt

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="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>
```

2.322.1 ConnectionProperties.ConnectString

The **ConnectionProperties.ConnectString** 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.ConnectString** element.

Parent elements
ConnectionProperties

The following is the XML Schema definition of the **ConnectionProperties.ConnectString** element.

```
<xsd:element name="ConnectString" type="xsd:string" />
```

2.322.2 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
ConnectionProperties

The following is the XML Schema definition of the **ConnectionProperties.DataProvider** element.

```
<xsd:element name="DataProvider" type="xsd:string" />
```

2.322.3 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
ConnectionProperties

The following is the XML Schema definition of the **ConnectionProperties.IntegratedSecurity** element.

```
<xsd:element name="IntegratedSecurity" type="xsd:boolean" minOccurs="0" />
```

2.322.4 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
ConnectionProperties

The following is the XML Schema definition of the **ConnectionProperties.Prompt** element.

```
<xsd:element name="Prompt" type="StringLocIDType" minOccurs="0" />
```

2.323 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
DataSets.DataSet

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>
```

2.323.1 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" />
```

2.324 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
DataSets

Attributes
DataSet.Name

Child elements
DataSet.AccentSensitivity
DataSet.CaseSensitivity
DataSet.Collation
DataSet.Fields
DataSet.Filters
DataSet.InterpretSubtotalsAsDetails
DataSet.KanatypeSensitivity
DataSet.Query
DataSet.SharedDataSet
DataSet.WidthSensitivity

Applies to RDL 2011/01

Child elements
DataSet.DefaultRelationships
DataSet.NullsAsBlanks
DataSet.CollationCulture

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: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: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>
```

2.324.1 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" />
```

2.324.2 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 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 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>
```

2.324.3 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>
```

2.324.4 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
DataSet

The following is the XML Schema definition of the **DataSet.Collation** element.

```
<xsd:element name="Collation" type="xsd:string" minOccurs="0" />
```

2.324.5 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" />
```

2.324.6 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" />
```

2.324.7 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>
```

2.324.8 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
DataSet

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>
```

2.324.9 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
DataSet

The following is the XML Schema definition of the **DataSet.Query** element.

```
<xsd:element name="Query" type="QueryType" minOccurs="0"/>
```

2.324.10 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
DataSet

The following is the XML Schema definition of the **DataSet.SharedDataSet** element

```
<xsd:element name="SharedDataSet" type="SharedDataSetType" minOccurs="0"/>
```

2.324.11 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>
```

2.324.12 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
DataSet

The following is the XML Schema definition of the **DataSet.DefaultRelationships** element.

```
<xsd:element name ="DefaultRelationships" type="DefaultRelationshipsType" minOccurs="0" />
```

2.324.13 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
DataSet

The following is the XML Schema definition of the **DataSet.NullsAsBlanks** element.

```
<xsd:element name ="NullsAsBlanks" type="xsd:boolean" minOccurs="0" />
```

2.324.14 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
DataSet

The following is the XML Schema definition of the **DataSet.CollationCulture** element.

```
<xsd:element name ="CollationCulture" type="xsd:string" minOccurs="0" />
```

2.325 SharedDataSet

The **SharedDataSet** element specifies information that is necessary to retrieve data for a DataSet.

The following are the parent and child elements of the **SharedDataSet** element.

Parent element
DataSet

Child elements
SharedDataSet.SharedDataSetReference
SharedDataSet.QueryParameters

The following is the XML Schema definition of the **SharedDataSet** element.

```
<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="skip" />
  </xsd:choice>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

2.325.1 SharedDataSet.SharedDataSetReference

The **SharedDataSet.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 **SharedDataSet.SharedDataSetReference** element.

Parent element
SharedDataSet

The following is the XML Schema definition of the **SharedDataSet.SharedDataSetReference** element.

```
<xsd:element name="SharedDataSetReference" type="xsd:string" />
```

2.325.2 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
SharedDataSet

The following is the XML Schema definition of the **SharedDataSet.QueryParameters** element.

```
<xsd:element name="QueryParameters" type="QueryParametersType" minOccurs="0" />
```

2.326 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
Fields.Field

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>
```

2.326.1 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" />
```

2.327 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
Fields

Attributes
Field.Name

Child elements
Field.DataField
Field.Value

Applies to RDL 2011/01

Child elements
Field.AggregateIndicatorField

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="StringWithDataTypeAttribute" 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>
```

2.327.1 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" />
```

2.327.2 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" />
```

2.327.3 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
Field.Value.DataType

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="StringWithDataTypeAttribute" minOccurs="0" />
```

2.327.4 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

2.327.5 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" />
```

2.328 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
Tablix
List
Matrix
Table
Group
Grouping
Chart
GaugePanel
CustomData
DataSet

Child elements
Filters.Filter

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>
```

2.328.1 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
Filters

The following is the XML Schema definition of the **Filters.Filter** element.

```
<xsd:element name="Filter" type="FilterType" maxOccurs="unbounded" />
```

2.329 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
Filters

Child elements
Filter.FilterExpression
Filter.FilterValues
Filter.Operator

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:anyAttribute namespace="##other" processContents="skip" />
  </xsd:complexType>
```

2.329.1 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
Filter

The following is the XML Schema definition of the **Filter.FilterExpression** element.

```
<xsd:element name="FilterExpression" type="xsd:string" />
```

2.329.2 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
Filter

The following is the XML Schema definition of the **Filter.FilterValues** element.

```
<xsd:element name="FilterValues" type="FilterValuesType" />
```

2.329.3 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.

Parent elements
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>
```

2.330 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.

Parent elements
Filter

Child elements
FilterValues.FilterValue

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="StringWithDataTypeAttribute"
     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="StringWithDataTypeAttribute"
     maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="lax" />
</xsd:complexType>
```

2.330.1 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
FilterValues

Attributes
FilterValues.FilterValue.DataType

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="StringWithDataTypeAttribute" maxOccurs="unbounded" />
```

2.330.2 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
FilterValues.FilterValue

2.331 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
DataSet

Child elements
Query.CommandText
Query.CommandType
Query.DataSourceName
Query.QueryParameters
Query.Timeout

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>
```

2.331.1 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
Query

The following is the XML Schema definition of the **Query.CommandText** element.

```
<xsd:element name="CommandText" type="xsd:string" />
```

2.331.2 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
Query

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>
```

2.331.3 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
Query

The following is the XML Schema definition of the **Query.DataSourceName** element.

```
<xsd:element name="DataSourceName" type="xsd:string" />
```

2.331.4 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
Query

The following is the XML Schema definition of the **Query.QueryParameters** element.

```
<xsd:element name="QueryParameters" type="QueryParametersType" minOccurs="0" />
```

2.331.5 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
Query

The following is the XML Schema definition of the **Query.Timeout** element.

```
<xsd:element name="Timeout" type="xsd:unsignedInt" minOccurs="0" />
```

2.332 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
Query
SharedDataSet

Child elements
QueryParameters.QueryParameter

The following is the XML Schema definition of the **QueryParameters** element.

```
<xsd:complexType name="QueryParametersType">
  <xsd:sequence>
    <xsd:element name="QueryParameter" type="QueryParameterType"
                 maxOccurs="unbounded" />
  </xsd:sequence>
  <xsd:anyAttribute namespace="##other" processContents="skip" />
</xsd:complexType>
```

2.332.1 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
QueryParameters

The following is the XML Schema definition of the **QueryParameters.QueryParameter** element.

```
<xsd:element name="QueryParameter" type="QueryParameterType"
             maxOccurs="unbounded" />
```

2.333 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
QueryParameters

Attributes
QueryParameter.Name

Child elements
QueryParameter.Value

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="StringWithDataTypeAttribute" />
    <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="StringWithDataTypeAttribute" />
    <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>
```

2.333.1 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
QueryParameter

The following is the XML Schema definition of the **QueryParameter.Name** attribute.

```
<xsd:attribute name="Name" type="xsd:string" use="required" />
```

2.333.2 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
QueryParameter

Attributes
QueryParameter.Value.DataType

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" />
```

2.333.3 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
QueryParameter.Value

2.334 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
DataSet

Child elements
DefaultRelationships.DefaultRelationship

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>
```

2.334.1 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.

Parent elements
DefaultRelationships

The following is the XML Schema definition of the **DefaultRelationships.DefaultRelationship** element.

```
<xsd:element name="DefaultRelationship" type="DefaultRelationshipType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.335 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.

Parent elements
DefaultRelationships

Child elements
DefaultRelationship.RelatedDataSet
DefaultRelationship.NaturalJoin
DefaultRelationship.JoinConditions

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>
```

2.335.1 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.

Parent elements
DefaultRelationship

The following is the XML Schema definition of the **DefaultRelationship.RelatedDataSet** element.

```
<xsd:element name="RelatedDataSet" type="xsd:string" />
```

2.335.2 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.

Parent elements
DefaultRelationship

The following is the XML Schema definition of the **DefaultRelationship.JoinConditions** element.

```
<xsd:element name="JoinConditions" type="JoinConditionsType" minOccurs="0" />
```

2.335.3 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.

Parent elements
DefaultRelationship

The following is the XML Schema definition of the **DefaultRelationship.NaturalJoin** element.

```
<xsd:element name="NaturalJoin" type="xsd:boolean" minOccurs="0" />
```

2.336 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.

Parent elements
DefaultRelationship

Child elements
JoinConditions.JoinCondition

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>
```

2.336.1 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.

Parent elements
JoinConditions

The following is the XML Schema definition of the **JoinConditions.JoinCondition** element.

```
<xsd:element name="JoinCondition" type="JoinConditionType" minOccurs="1"
             maxOccurs="unbounded" />
```

2.337 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.

Parent elements
JoinConditions

Child elements
JoinCondition.ForeignKey

Child elements
JoinCondition.PrimaryKey
JoinCondition.SortDirection

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:StringWithDataTypeAttribute" />
    <xsd:element name="PrimaryKey" type="rdl2010:StringWithDataTypeAttribute" />
    <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>
```

2.337.1 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>

Parent elements
JoinCondition

The following is the XML Schema definition of the **JoinCondition.ForeignKey** element.

```
<xsd:element name="ForeignKey" type="xsd:string" />
```

2.337.2 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.

Parent elements
JoinCondition

The following is the XML Schema definition of the **JoinCondition.PrimaryKey** element.

```
<xsd:element name="PrimaryKey" type="xsd:string" />
```

2.337.3 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.

Parent elements
JoinCondition

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>
```

2.338 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.

Parent elements
TablixCell

Parent elements
ChartDataPoint
DataCell

Child elements
Relationships.Relationship

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>
```

2.338.1 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
Relationships

The following is the XML Schema definition of the **Relationships.Relationship** element.

```
<xsd:element name ="Relationship" type="RelationshipType" />
```

2.339 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
Group
Relationships
Chart

Parent elements
MapDataRegion
GaugePanel
CustomData
Tablix

Child elements
Relationship.ParentScope
Relationship.NaturalJoin
Relationship.JoinConditions

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>
```

2.339.1 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
Relationship

The following is the XML Schema definition of the **Relationship.ParentScope** element.

```
<xsd:element name="ParentScope" type="xsd:string" minOccurs="0" />
```

2.339.2 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
Relationship

The following is the XML Schema definition of the **Relationship.NaturalJoin** element.

```
<xsd:element name="NaturalJoin" type="xsd:boolean" minOccurs="0" />
```

2.339.3 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
Relationship

The following is the XML Schema definition of the **Relationship.JoinConditions** element.

```
<xsd:element name="JoinConditions" type="JoinConditionsType" minOccurs="0" />
```

2.340 Expressions

2.340.1 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.

2.340.2 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 though 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(…)
```

2.340.3 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.

2.340.4 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>

2.340.5 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.

2.340.6 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
Fields	Fields in the current dataset.	Field
Parameters	Report parameters.	Parameter
ReportItems	All text boxes in the report.	**ReportItem**
Globals	Global variables.	**Variant**
User	User-specific data.	**Variant**

Applies to RDL 2012/01

Collection	Specification	Item type
Scopes	Scopes within the report.	Scope

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.

2.340.6.1 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
```

2.340.6.2 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
Value	**Variant**	
IsMissing	**Boolean**	
UniqueName	String ([XMLSCHEMA2/2] section 3.2.1)	
BackgroundColor	**String**	See Style.BackgroundColor
Color	**String**	See Style.Color
FontFamily	**String**	See Style.FontFamily
FontSize	**String**	See Style.FontSize
FontWeight	**String**	See Style.FontWeight
FontStyle	**String**	See Style.FontStyle
TextDecoration	**String**	See Style.TextDecoration
FormattedValue	**String**	
Key	**Variant**	
LevelNumber	**Integer**	
ParentUniqueName	**String**	

2.340.6.3 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.

2.340.6.4 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
Value	**Variant** **VariantArray**	The **Value** property is determined by the parameter value input provided to the current report execution session. The data type of **Value** is a **Variant** for single-value parameters and **VariantArray** for multivalued parameters.
Label	String ([XMLSCHEMA2/2] section 3.2.1) **StringArray**	The **Label** property is determined by the value of the **Label** or **LabelField** element of the corresponding parameter value in the **ValidValues** list. The data type of **Label** is a **String** for single-value parameters and **StringArray** for multivalued parameters.
Count	Integer ([XMLSCHEMA2/2] section 3.3.17)	The **Count** property specifies the number of values and labels. The value of the **Count** 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).
IsMultiValue	Boolean ([XMLSCHEMA2/2] section 3.2.2)	The **IsMultiValue** property specifies whether the parameter is multivalued. The value of this property is **Boolean**.

2.340.6.5 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.

2.340.6.6 Globals

Members of the **Globals** collection are of type **Variant**, but individual members have the following known return types.

Property name	Type	Specification
PageNumber	Integer ([XMLSCHEMA2/2] section 3.3.17)	Current page number. The member MUST NOT be referenced outside of the page header or page footer of the report.
TotalPages	**Integer**	Total number of pages in the report. The member MUST NOT be referenced outside of the page header or page footer of the report.
ExecutionTime	DateTime ([XMLSCHEMA2] section 3.2.7)	The date and time at which the report began executing.
ReportServerUrl	String ([XMLSCHEMA2/2] section 3.2.1)	URL to the report server, such as http://reportserver/reports.
ReportFolder	**String**	Full path on the report server to the folder containing the report. For example, for the report http://reportserver/reports/salesreports/budgeting/currentbudget, the **ReportFolder** is /salesreports/budgeting.
ReportName	**String**	Name of the report in the report catalog. For example, for the report http://reportserver/reports/salesreports/budgeting/currentbudget, the **ReportName** is "currentbudget".
RenderFormat.Name	**String**	Name of the renderer.<191>
RenderFormat.IsInteractive	Boolean ([XMLSCHEMA2/2] section 3.2.2)	Specifies whether the current user request is for an interactive rendering format.
RenderFormat.DeviceInfo	ReadOnlyNameValueCollection	A collection that holds the keys and values of the **deviceinfo** parameters of the current rendering request.<192>

2.340.6.7 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.

2.340.6.8 User

Members of the **User** collection are of type **Variant**, but individual members have the following known return types.

Property name	Type	Specification
UserID	String ([XMLSCHEMA2/2] section 3.2.1)	ID of the user executing the report.
Language	**String**	Language ID of the client executing the report.

2.340.6.9 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
Type	String ([XMLSCHEMA2/2] section 3.2.1)	Type of **data provider** for the data source.
DataSourceReference	**String**	Path to the data source (**Nothing** for embedded data sources).

2.340.6.10 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
CommandText	String ([XMLSCHEMA2/2] section 3.2.1)	The **CommandText** of the dataset.
RewrittenCommandText	**String**	The **CommandText** of the dataset after optionally rewritten by the report execution engine.

2.340.6.11 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
SetValue(Serializable)	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).

2.340.6.12 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<193>	No	No
Query Parameter	No	No	Yes	No	No	No	No
Group Expression	Yes	No	Yes	No	Yes	No	No

Usage	Global collections						
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.

2.340.6.13 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
Fields	Fields	The **Fields** from the **DataSet** 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
```

2.340.7 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.

2.340.7.1 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.

2.340.7.2 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

2.340.7.3 Recursive Depth

In recursive hierarchies, the **Level** function can be used to determine the current depth of the recursive hierarchy.

2.340.7.4 Function: Level

The following are the arguments, types, and specification of the **Level** function.

Function	Arguments	Type	Specification
Level	*Return*	**Integer**	A zero-based **Integer** 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, **Level** returns 0.
	Scope	**String**	Optional. Defaults to the current scope.

2.340.7.5 Aggregate Function: Sum

The following are the arguments, types, and specification of the **Sum** aggregate function.

Function	Arguments	Type	Specification
Sum	*Return*	Float	Returns the sum of all values of the expression within the scope. *Return* type is decimal for decimal* expressions and double for all other

Function	Arguments	Type	Specification
			expressions.
	Expression	**Numeric**	The expression to aggregate. MUST NOT contain the **First**, **Last**, **Previous**, or **RunningValue** functions. MUST NOT reference other aggregate functions unless their *Scope* is the same as, or a child scope of, this function's *Scope*. 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 *Scope* parameter MUST NOT contain any aggregate functions. Aggregate functions with the name of a dataset as its *Scope* parameter MUST NOT contain any aggregate functions.
	Scope	String	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 Scope.
	Recursive	**Enum**	Recursive | Simple (Default). Specifies whether the aggregate is calculated recursively. Optional. See also Recursive.

* 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.

2.340.7.6 Aggregate Function: Avg

The following are the arguments, types, and specification of the **Avg** aggregate function.

Function	Arguments	Type	Specification
Avg	*Return*	Float	Returns the average of all non-null values of the expression within the specified scope. For the *Return* type, see the Sum function.
	<Expression>	**Numeric**	See the **Sum** function.
	Scope	String	See the **Sum** function.
	Recursive	**Enum**	See the **Sum** function.

2.340.7.7 Aggregate Function: Max

The following are the arguments, types, and specification of the **Max** aggregate function.

Function	Arguments	Type	Specification
Max	*Return*	**Variant**	Returns the maximum of all non-null values of the expression within the specified scope. *Return* type is the same as the expression type.
	Expression	**Variant**	See the Sum function.

Function	Arguments	Type	Specification
	Scope	String	See the **Sum** function.
	Recursive	**Enum**	See the **Sum** function.

2.340.7.8 Aggregate Function: Min

The following are the arguments, types, and specification of the **Min** aggregate function.

Function	Arguments	Type	Specification
Min	*Return*	**Variant**	Returns the minimum of all non-null values of the expression within the specified scope. *Return* type is the same as the expression type.
	Expression	**Variant**	See the Sum function.
	Scope	String	See the **Sum** function.
	Recursive	**Enum**	See the **Sum** function.

2.340.7.9 Aggregate Function: Count

The following are the arguments, types, and specification of the **Count** aggregate function.

Function	Arguments	Type	Specification
Count	*Return*	Integer	Returns the count of all non-null values of the expression within the specified scope.
	Expression	**Variant** or **Binary**	See the Sum function.
	Scope	String	See the **Sum** function.
	Recursive	**Enum**	See the **Sum** function.

2.340.7.10 Aggregate Function: CountDistinct

The following are the arguments, types, and specification of the **CountDistinct** aggregate function.

Function	Arguments	Type	Specification
CountDistinct	*Return*	Integer	Returns the count of all distinct non-null values of the expression within the specified scope.
	Expression	**Variant**	See the Sum function**.**
	Scope	String	See the **Sum** function.

Function	Arguments	Type	Specification
	Recursive	**Enum**	See the **Sum** function.

2.340.7.11 Aggregate Function: CountRows

The following are the arguments, types, and specification of the **CountRows** aggregate function.

Function	Arguments	Type	Specification
CountRows	*Return*	Integer	Returns the count of all rows within the specified scope.
	Scope	String	See the Sum function.
	Recursive	**Enum**	See the **Sum** function.

2.340.7.12 Aggregate Function: StDev

The following are the arguments, types, and specification of the **StDev** aggregate function.

Function	Arguments	Type	Specification
StDev	*Return*	Float	Returns the standard deviation of all non-null values of the expression within the specified scope.
	Expression	**Numeric**	See the Sum function.
	Scope	String	See the **Sum** function.
	Recursive	**Enum**	See Sum.

2.340.7.13 Aggregate Function: StDevP

The following are the arguments, types, and specification of the **StDevP** aggregate function.

Function	Arguments	Type	Specification
StDevP	*Return*	Float	Returns the population standard deviation of all non-null values of the expression within the specified scope. *Return* type is the same as the expression type.
	Expression	**Numeric**	See the Sum function.
	Scope	String	See the **Sum** function.
	Recursive	**Enum**	See the **Sum** function.

2.340.7.14 Aggregate Function: Var

The following are the arguments, types, and specification of the **Var** aggregate function.

Function	Arguments	Type	Specification
Var	*Return*	Float	Returns the variance of all non-null values of the expression within the specified scope. See the Sum function regarding the *return* type.
	Expression	**Numeric**	See the **Sum** function.
	Scope	String	See the **Sum** function.
	Recursive	**Enum**	See the **Sum** function.

2.340.7.15 Aggregate Function: VarP

The following are the arguments, types, and specification of the **VarP** aggregate function.

Function	Arguments	Type	Specification
VarP	*Return*	Float	Returns the population variance of all non-null values of the expression within the specified scope. See the Sum function regarding the *return* type.
	Expression	**Numeric**	See the **Sum** function.
	Scope	String	See the **Sum** function.
	Recursive	**Enum**	See the **Sum** function.

2.340.7.16 Aggregate Function: Union

The following are the arguments, types, and specification of the **Union** aggregate function.

Function	Arguments	Type	Specification
Union	*Return*	SqlGeography or SqlGeometry	Returns the union of all values of the expression within the scope. *Return* type is SqlGeography for SqlGeography expressions and SqlGeometry for SqlGeometry expressions.
	Expression	SqlGeography or SqlGeometry	See the Sum function. All the expressions MUST be either of type SqlGeography or SqlGeometry.
	Scope	String	See the **Sum** function.

2.340.7.17 Aggregate Function: First

The following are the arguments, types, and specification of the **First** aggregate function.

Function	Arguments	Type	Specification
First	*Return*	**Variant** or **Binary**	Returns the first value of the expression within the scope (after all sorting up through the *Scope* has been applied). *Return* type is the same as the expression type.
	Expression	**Variant** or **Binary**	See the Sum function.
	Scope	String	See the **Sum** function.

2.340.7.18 Aggregate Function: Last

The following are the arguments, types, and specification of the **Last** aggregate function.

Function	Arguments	Type	Specification
Last	*Return*	**Variant** or **Binary**	Returns the last value of the expression within the scope (after all sorting up through the *Scope* has been applied). *Return* type is the same as the expression type.
	Expression	**Variant** or **Binary**	See the Sum function.
	Scope	String	See the **Sum** function.

2.340.7.19 Aggregate Function: Previous

The following are the arguments, types, and specification of the **Previous** aggregate function.

Function	Arguments	Type	Specification
Previous	*Return*	**Variant** or **Binary**	Returns the value of the expression for the previous instance of *PreviousScope*, or (if the expression is an aggregate), returns the value of the aggregate expression as applied to the previous instance of the *PreviousScope* corresponding to the current instance of the *Scope* of the aggregate function. Returns NULL if there is no corresponding previous instance.
	Expression	**Variant** or **Binary**	The expression for which to retrieve the previous value. If the expression contains an aggregate, **Previous** aggregates the data within the previous instance of the *PreviousScope* that corresponds to the current instance of the aggregate's *Scope*. An aggregate *Scope* equal to the *PreviousScope* indicates that all of the data in the *PreviousScope* is aggregated. The scope of the aggregate MUST be contained by (or equal to) *PreviousScope*. The aggregate function MUST NOT be **Aggregate** or **Previous**. The aggregate MUST NOT be recursive. The functions Level and InScope MUST NOT be used in the expression.
	PreviousScope	String	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. **Previous** retrieves the data in the previous instance of the group or data region. A *PreviousScope* of **Nothing** specifies that **Previous** retrieves the value of the expression for the previous detail

Function	Arguments	Type	Specification
			row of data. *PreviousScope* MUST be a constant, not an expression. Optional. Default: The current scope (**Nothing** if in a **detail scope**).

2.340.7.20 Aggregate Function: RunningValue

The following are the arguments, types, and specification of the **RunningValue** aggregate function.

Function	Argument	Type	Specification
RunningValue	*Return*	See the *Function* argument.*	A running aggregate of the expression, using the specified aggregate function.
	Expression	See *Function*	The expression to aggregate. MUST NOT contain the **First**, **Last**, **Previous**, or **RunningValue** functions. Can reference other aggregate functions if their *Scope* is the same as, or a child scope of, this function's *Scope*. For all distinct scopes directly** referenced in the expression, there MUST be one scope that has all other scopes as its ancestor.
	Function	**Enum**	Name of an aggregate function for which to calculate a running value (MUST NOT be **CountRows**, **RunningValue**, **RowNumber**, or **Aggregate**). *Expression* type and *Return* type are determined by the aggregate function used.
	Scope	String	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 *Scope* argument MUST be a constant, not an expression.

* The *Return* data type is determined by the function that is used in the *Function***Enum** argument of the **RunningValue** call. For example, if **Sum** is used as *Function***Enum** 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.

2.340.7.21 Aggregate Function: RowNumber

The following are the arguments, types, and specification of the **RowNumber** aggregate function.

Function	Arguments	Type	Specification
RowNumber	*Return*	Integer	The row number of the current row or group instance.
	Scope	String	See RunningValue.

2.340.7.22 Aggregate Function: Aggregate

The following are the arguments, types, and specification of the **Aggregate** aggregate function.

Function	Arguments	Type	Specification
Aggregate	*Return*	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.
	Expression	**Variant**	The expression to aggregate. MUST be a simple field reference, for example: =Aggregate(Fields!Sales.Value,Year)
	Scope	String	See the Sum function. All group expressions for the *Scope* specified (and all containing group scopes) MUST be simple field references or (non-expression) constants.

2.340.7.23 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**.

2.340.7.24 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.

	Aggregation / Special Function Call							
Usage Context	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.

2.340.7.25 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.

2.340.7.26 Additional Functions

This section describes additional functions that can be referenced from expressions.

2.340.7.26.1 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.

2.340.7.26.2 Function: InScope

Function	Arguments	Type	Specification
InScope	*Return*	Boolean	Returns true if the current instance is within the specified scope. Otherwise, returns false.
	Scope	String	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>
```

2.340.7.26.3 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.

2.340.7.26.4 Function: CreateDrillthroughContext

Function	Arguments	Type	Specification
CreateDrillthroughContext	*Return*	String	Returns a **DrillthroughContext** parameter value that specifies the current drillthrough context, including semantic query fields from the current scope referenced in the **Value** 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.

2.340.7.26.5 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
Lookup	*Return*	**Variant**	**Lookup** evaluates the *Source* expression for the current instance of the current scope. It then finds the first row of data in the specified *Dataset** for which the source value equals the *Destination* expression evaluated for that row.** It returns the value of the *Result* expression evaluated for that row, or NULL if no matching row is found. The *Return* data type is the same as the *Result* data type.
	Source	**Variant**	Specifies the value to look up. The *Source* expression result MUST match the data type of the *Destination* expression. The expression MUST NOT reference lookup functions.***
	Destination	**Variant**	Specifies the value to compare the *Source* expression to. The *Destination* expression MUST NOT reference aggregate functions, report items, or lookup functions.
	Result	**Variant**	The *Result* expression MUST NOT reference aggregate functions, report items, or lookup functions.
	Dataset	String	Name of the dataset in which to evaluate the *Destination* expression and *Result* 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**).

2.340.7.26.6 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
LookupSet	*Return*	**VariantArray**	**LookupSet** evaluates the *Source* expression for the current instance of the current scope. It then finds all rows of data in the specified dataset for which the source value equals the *Destination* expression evaluated for that row. It returns an array containing a list of all values of the *Result* 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. Elements of the *Return* array MUST be the same data type as the *Result* data type.
	Source	**Variant**	Specifies the value to look up. The value of the expression result MUST match the data type of the *Destination* expression. The expression MUST NOT reference lookup functions.
	Destination	**Variant**	Specifies the value to compare the *Source* expression to. The *Destination* expression MUST NOT reference aggregate functions, report item, or lookup functions.
	Result	**Variant**	The *Result* expression MUST NOT reference aggregate functions, report items, or lookup functions.
	DataSet	String ([XMLSCHEMA2/2] section 3.2.1)	Name of the dataset in which to evaluate the *Destination* expression and *Result* expression. MUST be a constant, not an expression.

2.340.7.26.7 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
MultiLookup	*Return*	**VariantArray**	**MultiLookup** evaluates the *Source* expression for the current instance of the current scope. Then, for each item in the *Source* array, it finds the first row of data in the specified dataset for which the item in the *Source* array equals the *Destination* expression evaluated for that row.<194> It returns a **Variant** array containing a list of values of the *Result* expression evaluated for each such first matching row. **MultiLookup** 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 *Source* array. Elements of the *Return* array MUST be the same data type as the *Result* type.
	Source	**VariantArray**<195>	The value to look up. Items in the array MUST match the data type of the *Destination* expression. The expression MUST NOT reference lookup functions.*

Function	Arguments	Type	Description
	Destination	**Variant**	Specifies the value to compare the *Source* expression to. The *Destination* expression MUST NOT reference aggregate functions, report items, or lookup functions.*
	Result	**Variant**	The expression MUST NOT reference aggregate functions, report items, or lookup functions.*
	Dataset	**String**	Name of the dataset in which to evaluate the *Destination* expression and *Result* expression. MUST be a constant, not an expression.

* "Lookup functions" refers to all types of lookup functions (such as Lookup, LookupSet, and **MultiLookup**).

2.340.7.26.8 Function: MinValue

Applies to RDL 2011/01

The following are the arguments and types of the **MinValue** function.

Function	Arguments	Type	Description
MinValue	*Return*	**Variant**	**MinValue** accepts a variable number of arguments and returns the smallest argument value. **MinValue** MUST NOT accept less than two arguments. If any argument contains an evaluation error, **MinValue** produces an evaluation error. The data type of the *Return* value MUST be the same as the data type of the smallest argument value.
	Arg1	**Variant**	The first value to compare.
	Arg2	**Variant**	The second value to compare.
	ArgN	**Variant**	The Nth value to compare.

2.340.7.26.9 Function: MaxValue

Applies to RDL 2011/01

The following are the arguments and types of the **MaxValue** function.

Function	Arguments	Type	Description
MaxValue	*Return*	**Variant**	**MaxValue** accepts a variable number of arguments and returns the largest argument value. **MaxValue** MUST NOT accept less than two arguments. If any argument contains an evaluation error, **MaxValue** produces an evaluation error. The data type of the *Return* value MUST be the same as the data type of the smallest argument value.
	Arg1	**Variant**	The first value to compare.
	Arg2	**Variant**	The second value to compare.
	ArgN	**Variant**	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.

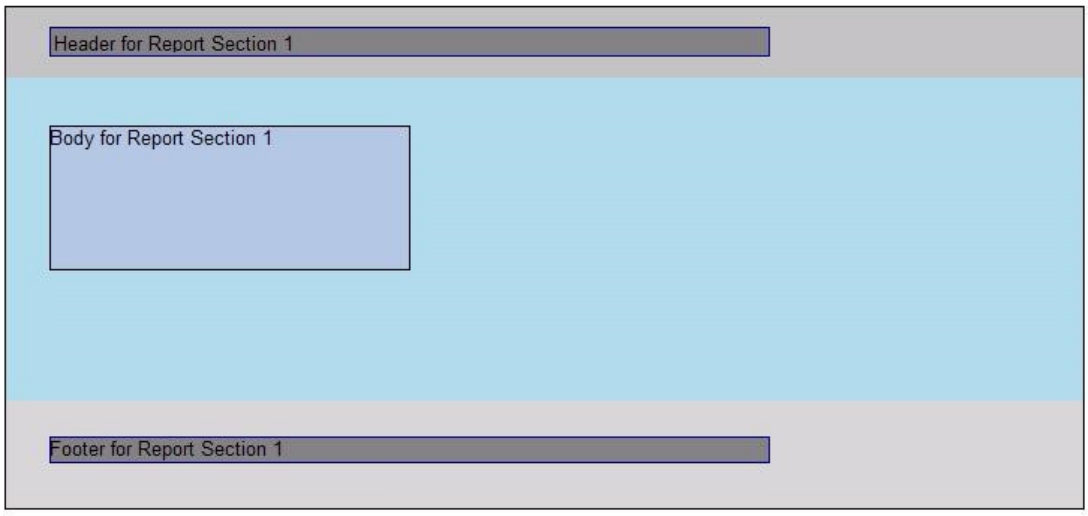

Figure 9: RDL schema 2008/01 report rendering

```
<?xml version="1.0" encoding="utf-8"?>
<Report xmlns="http://schemas.microsoft.com/sqlserver/reporting/2008/01/reportdefinition">
 <DataSources>
   <DataSource Name="DataSource1">
     <ConnectionProperties>
       <DataProvider>SQL</DataProvider>
       <ConnectString>data source=DataServer; initial catalog=Northwind;</ConnectString>
     </ConnectionProperties>
   </DataSource>
 </DataSources>
 <DataSets>
   <DataSet Name="DataSet1">
     <Fields>
       <Field Name="ProductID">
         <DataField>ProductID</DataField>
       </Field>
       <Field Name="ProductName">
         <DataField>ProductName</DataField>
       </Field>
       <Field Name="SupplierID">
         <DataField>SupplierID</DataField>
       </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        <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>
```

```
  </PageFooter>
  <PageWidth>8.5in</PageWidth>
  <PageHeight>4in</PageHeight>
  <InteractiveWidth>8.5in</InteractiveWidth>
  <LeftMargin>0.5in</LeftMargin>
  <RightMargin>0.5in</RightMargin>
  <TopMargin>0.25in</TopMargin>
  <BottomMargin>0.25in</BottomMargin>
  <Style>
    <Border>
      <Color>Black</Color>
      <Style>Solid</Style>
      <Width>1pt</Width>
    </Border>
  </Style>
</Page>
<Width>6.5in</Width>
<Language>=User!Language</Language>
<Variables>
  <Variable Name="ReportVariable1">
    <Value>=1</Value>
  </Variable>
</Variables>
<ConsumeContainerWhitespace>true</ConsumeContainerWhitespace>
<ReportParameters>
  <ReportParameter Name="Product">
    <DataType>String</DataType>
    <DefaultValue>
      <DataSetReference>
        <DataSetName>DataSet1</DataSetName>
        <ValueField>ProductID</ValueField>
      </DataSetReference>
    </DefaultValue>
    <Prompt>Product</Prompt>
    <ValidValues>
      <DataSetReference>
        <DataSetName>DataSet1</DataSetName>
        <ValueField>ProductID</ValueField>
        <LabelField>ProductName</LabelField>
      </DataSetReference>
    </ValidValues>
    <MultiValue>true</MultiValue>
  </ReportParameter>
  <ReportParameter Name="Quantity">
    <DataType>Integer</DataType>
    <DefaultValue>
      <Values>
        <Value DataType="Integer">0</Value>
      </Values>
    </DefaultValue>
    <Prompt>Quantity</Prompt>
    <ValidValues>
      <ParameterValues>
        <ParameterValue>
          <Value DataType="Integer">0</Value>
          <Label>0</Label>
        </ParameterValue>
        <ParameterValue>
          <Value DataType="Integer">10</Value>
          <Label>10</Label>
        </ParameterValue>
        <ParameterValue>
          <Value DataType="Integer">50</Value>
          <Label>50</Label>
        </ParameterValue>
      </ParameterValues>
    </ValidValues>
  </ReportParameter>
</ReportParameters>
```

```
228</Report>
```

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.

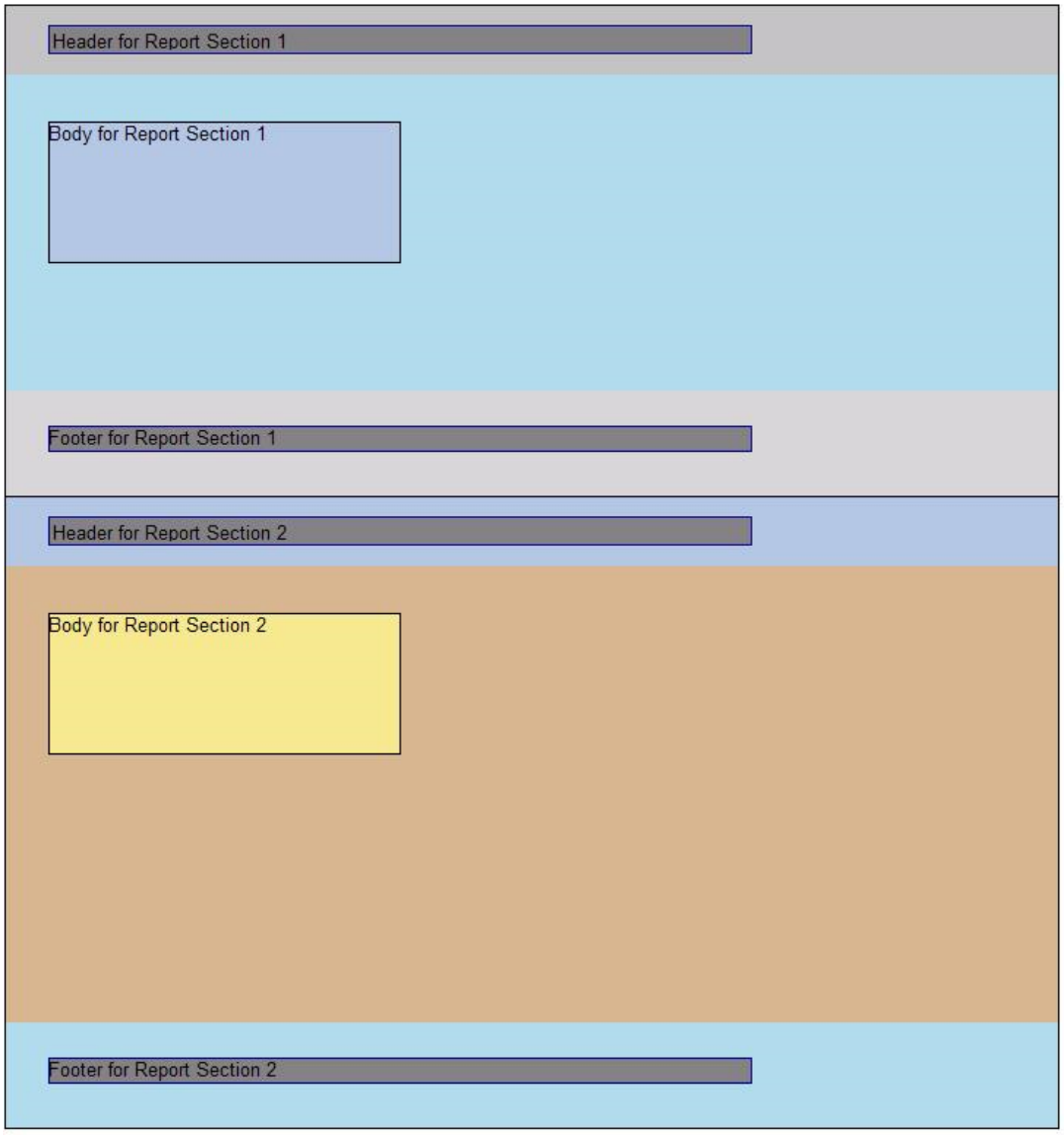

Figure 10: RDL schema 2010/01 report rendering

```
<?xml version="1.0" encoding="utf-8"?>
<Report
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2010/01/reportdefinition">
 <AutoRefresh>3600</AutoRefresh>
 <DataSources>
   <DataSource Name="DataSource1">
     <ConnectionProperties>
       <DataProvider>SQL</DataProvider>
       <ConnectString>data source=DataServer; initial
          catalog=northwind;</ConnectString>
     </ConnectionProperties>
   </DataSource>
 </DataSources>
 <DataSets>
   <DataSet Name="DataSet1">
     <Fields>
       <Field Name="ProductID">
         <DataField>ProductID</DataField>
       </Field>
       <Field Name="ProductName">
         <DataField>ProductName</DataField>
       </Field>
       <Field Name="SupplierID">
         <DataField>SupplierID</DataField>
       </Field>
       <Field Name="CategoryID">
         <DataField>CategoryID</DataField>
       </Field>
       <Field Name="QuantityPerUnit">
         <DataField>QuantityPerUnit</DataField>
       </Field>
       <Field Name="UnitPrice">
         <DataField>UnitPrice</DataField>
       </Field>
       <Field Name="UnitsInStock">
         <DataField>UnitsInStock</DataField>
       </Field>
       <Field Name="UnitsOnOrder">
         <DataField>UnitsOnOrder</DataField>
       </Field>
       <Field Name="ReorderLevel">
         <DataField>ReorderLevel</DataField>
       </Field>
       <Field Name="Discontinued">
         <DataField>Discontinued</DataField>
       </Field>
     </Fields>
     <Query>
       <DataSourceName>DataSource1</DataSourceName>
       <CommandText>Select * From Products</CommandText>
     </Query>
   </DataSet>
 </DataSets>
 <ReportSections>
   <ReportSection>
     <Body>
       <ReportItems>
         <Textbox Name="BodyTextbox1">
           <CanGrow>true</CanGrow>
           <KeepTogether>true</KeepTogether>
           <Paragraphs>
             <Paragraph>
               <TextRuns>
                 <TextRun>
                   <Value>Body for Report Section 1</Value>
                 </TextRun>
               </TextRuns>
             </Paragraph>
           </Paragraphs>
```

```
            <Top>0.33333in</Top>
            <Left>0.3125in</Left>
            <Height>1in</Height>
            <Width>2.5in</Width>
            <Style>
              <Border>
                <Color>Black</Color>
                <Style>Solid</Style>
                <Width>1pt</Width>
              </Border>
              <BackgroundColor>LightSteelBlue</BackgroundColor>
            </Style>
          </Textbox>
        </ReportItems>
        <Height>2.5in</Height>
        <Style>
          <BackgroundColor>LightBlue</BackgroundColor>
        </Style>
      </Body>
      <Width>6.5in</Width>
      <Page>
        <PageHeader>
          <Height>0.5in</Height>
          <PrintOnFirstPage>true</PrintOnFirstPage>
          <PrintOnLastPage>true</PrintOnLastPage>
          <PrintBetweenSections>true</PrintBetweenSections>
          <ReportItems>
            <Textbox Name="HeaderTextbox1">
              <Paragraphs>
                <Paragraph>
                  <TextRuns>
                    <TextRun>
                       <Value>Header for Report Section 1</Value>
                     </TextRun>
                  </TextRuns>
                </Paragraph>
              </Paragraphs>
              <Top>0.14583in</Top>
              <Left>0.3125in</Left>
              <Height>0.19792in</Height>
              <Width>5in</Width>
              <Style>
                <Border>
                  <Color>DarkBlue</Color>
                  <Style>Solid</Style>
                  <Width>1pt</Width>
                </Border>
                <BackgroundColor>Gray</BackgroundColor>
                <PaddingLeft>2pt</PaddingLeft>
                <PaddingRight>2pt</PaddingRight>
                <PaddingTop>2pt</PaddingTop>
                <PaddingBottom>2pt</PaddingBottom>
              </Style>
            </Textbox>
          </ReportItems>
          <Style>
            <BackgroundColor>Silver</BackgroundColor>
          </Style>
        </PageHeader>
        <PageFooter>
          <Height>0.75in</Height>
          <PrintOnFirstPage>true</PrintOnFirstPage>
          <PrintOnLastPage>true</PrintOnLastPage>
          <PrintBetweenSections>true</PrintBetweenSections>
          <ReportItems>
            <Textbox Name="FooterTextbox1">
              <CanGrow>true</CanGrow>
              <KeepTogether>true</KeepTogether>
              <Paragraphs>
```

```
              <Paragraph>
                <TextRuns>
                  <TextRun>
                    <Value>Footer for Report Section 1</Value>
                  </TextRun>
                </TextRuns>
              </Paragraph>
            </Paragraphs>
            <Top>0.25in</Top>
            <Left>0.3125in</Left>
            <Height>0.17708in</Height>
            <Width>5in</Width>
            <Style>
              <Border>
                <Color>DarkBlue</Color>
                <Style>Solid</Style>
                <Width>1pt</Width>
              </Border>
              <BackgroundColor>Gray</BackgroundColor>
            </Style>
          </Textbox>
        </ReportItems>
        <Style>
          <BackgroundColor>LightGrey</BackgroundColor>
        </Style>
      </PageFooter>
      <PageWidth>8.5in</PageWidth>
      <PageHeight>5in</PageHeight>
      <InteractiveWidth>8.5in</InteractiveWidth>
      <LeftMargin>0.5in</LeftMargin>
      <RightMargin>0.5in</RightMargin>
      <TopMargin>0.25in</TopMargin>
      <BottomMargin>0.25in</BottomMargin>
      <Style>
        <Border>
          <Color>Black</Color>
          <Style>Solid</Style>
          <Width>1pt</Width>
        </Border>
      </Style>
    </Page>
  </ReportSection>
  <ReportSection>
    <Body>
      <ReportItems>
        <Textbox Name="BodyTextbox2">
          <CanGrow>true</CanGrow>
          <KeepTogether>true</KeepTogether>
          <Paragraphs>
            <Paragraph>
              <TextRuns>
                <TextRun>
                  <Value>Body for Report Section 2</Value>
                </TextRun>
              </TextRuns>
            </Paragraph>
          </Paragraphs>
          <Top>0.33333in</Top>
          <Left>0.3125in</Left>
          <Height>1in</Height>
          <Width>2.5in</Width>
          <Style>
            <Border>
              <Color>Black</Color>
              <Style>Solid</Style>
              <Width>1pt</Width>
            </Border>
            <BackgroundColor>Khaki</BackgroundColor>
          </Style>
```

```
        </Textbox>
      </ReportItems>
      <Height>3in</Height>
      <Style>
        <BackgroundColor>Tan</BackgroundColor>
      </Style>
    </Body>
    <Width>6.5in</Width>
    <Page>
      <PageHeader>
        <Height>0.5in</Height>
        <PrintOnFirstPage>true</PrintOnFirstPage>
        <PrintOnLastPage>true</PrintOnLastPage>
        <PrintBetweenSections>true</PrintBetweenSections>
        <ReportItems>
          <Textbox Name="HeaderTextbox2">
            <Paragraphs>
              <Paragraph>
                <TextRuns>
                  <TextRun>
                    <Value>Header for Report Section 2</Value>
                  </TextRun>
                </TextRuns>
              </Paragraph>
            </Paragraphs>
            <Top>0.14583in</Top>
            <Left>0.3125in</Left>
            <Height>0.19792in</Height>
            <Width>5in</Width>
            <Style>
              <Border>
                <Color>DarkBlue</Color>
                <Style>Solid</Style>
                <Width>1pt</Width>
              </Border>
              <BackgroundColor>Gray</BackgroundColor>
              <PaddingLeft>2pt</PaddingLeft>
              <PaddingRight>2pt</PaddingRight>
              <PaddingTop>2pt</PaddingTop>
              <PaddingBottom>2pt</PaddingBottom>
            </Style>
          </Textbox>
        </ReportItems>
        <Style>
          <BackgroundColor>LightSteelBlue</BackgroundColor>
        </Style>
      </PageHeader>
      <PageFooter>
        <Height>0.75in</Height>
        <PrintOnFirstPage>true</PrintOnFirstPage>
        <PrintOnLastPage>true</PrintOnLastPage>
        <PrintBetweenSections>true</PrintBetweenSections>
        <ReportItems>
          <Textbox Name="FooterTextbox2">
            <CanGrow>true</CanGrow>
            <KeepTogether>true</KeepTogether>
            <Paragraphs>
              <Paragraph>
                <TextRuns>
                  <TextRun>
                    <Value>Footer for Report Section 2</Value>
                  </TextRun>
                </TextRuns>
              </Paragraph>
            </Paragraphs>
            <Top>0.25in</Top>
            <Left>0.3125in</Left>
            <Height>0.17708in</Height>
            <Width>5in</Width>
```

```
            <Style>
              <Border>
                <Color>DarkBlue</Color>
                <Style>Solid</Style>
                <Width>1pt</Width>
              </Border>
              <BackgroundColor>Gray</BackgroundColor>
            </Style>
          </Textbox>
        </ReportItems>
        <Style>
          <BackgroundColor>LightBlue</BackgroundColor>
        </Style>
      </PageFooter>
      <PageWidth>8.5in</PageWidth>
      <PageHeight>5in</PageHeight>
      <InteractiveWidth>8.5in</InteractiveWidth>
      <LeftMargin>0.5in</LeftMargin>
      <RightMargin>0.5in</RightMargin>
      <TopMargin>0.25in</TopMargin>
      <BottomMargin>0.25in</BottomMargin>
      <Style>
        <Border>
          <Color>Black</Color>
          <Style>Solid</Style>
          <Width>1pt</Width>
        </Border>
      </Style>
    </Page>
  </ReportSection>
</ReportSections>
<Language>=User!Language</Language>
<Variables>
  <Variable Name="ReportVariable1">
    <Value>=1</Value>
  </Variable>
</Variables>
<ConsumeContainerWhitespace>true</ConsumeContainerWhitespace>
<ReportParameters>
  <ReportParameter Name="Product">
    <DataType>String</DataType>
    <DefaultValue>
      <DataSetReference>
        <DataSetName>DataSet1</DataSetName>
        <ValueField>ProductID</ValueField>
      </DataSetReference>
    </DefaultValue>
    <Prompt>Product</Prompt>
    <ValidValues>
      <DataSetReference>
        <DataSetName>DataSet1</DataSetName>
        <ValueField>ProductID</ValueField>
        <LabelField>ProductName</LabelField>
      </DataSetReference>
    </ValidValues>
    <MultiValue>true</MultiValue>
  </ReportParameter>
  <ReportParameter Name="Quantity">
    <DataType>Integer</DataType>
    <DefaultValue>
      <Values>
        <Value DataType="Integer">0</Value>
      </Values>
    </DefaultValue>
    <Prompt>Quantity</Prompt>
    <ValidValues>
      <ParameterValues>
        <ParameterValue>
          <Value DataType="Integer">0</Value>
```

```
            <Label>0</Label>
          </ParameterValue>
          <ParameterValue>
            <Value DataType="Integer">10</Value>
            <Label>10</Label>
          </ParameterValue>
          <ParameterValue>
            <Value DataType="Integer">50</Value>
            <Label>50</Label>
          </ParameterValue>
        </ParameterValues>
      </ValidValues>
    </ReportParameter>
  </ReportParameters>
</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.

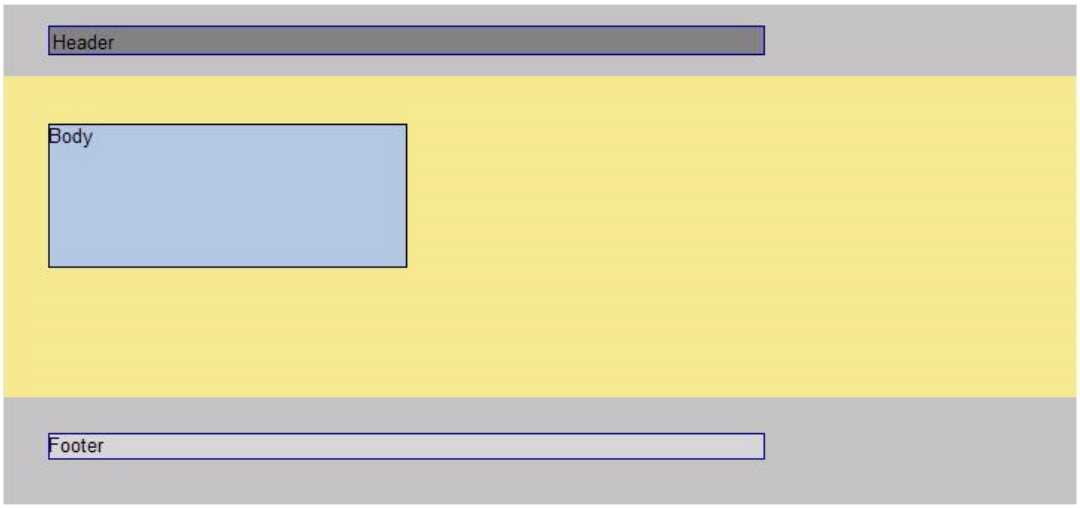

Figure 11: RDL schema 2005/01 report rendering

```
<?xml version="1.0" encoding="utf-8"?>
<Report xmlns="http://schemas.microsoft.com/sqlserver/reporting/2005/01/reportdefinition">
  <DataSources>
    <DataSource Name="DataSource1">
      <ConnectionProperties>
        <DataProvider>SQL</DataProvider>
        <ConnectString>data source=DataServer; initial catalog=Northwind;</ConnectString>
      </ConnectionProperties>
    </DataSource>
  </DataSources>
  <DataSets>
    <DataSet Name="DataSet1">
      <Fields>
```

```
        <Field Name="ProductID">
          <DataField>ProductID</DataField>
        </Field>
        <Field Name="ProductName">
          <DataField>ProductName</DataField>
        </Field>
        <Field Name="SupplierID">
          <DataField>SupplierID</DataField>
        </Field>
        <Field Name="CategoryID">
          <DataField>CategoryID</DataField>
        </Field>
        <Field Name="QuantityPerUnit">
          <DataField>QuantityPerUnit</DataField>
        </Field>
        <Field Name="UnitPrice">
          <DataField>UnitPrice</DataField>
        </Field>
        <Field Name="UnitsInStock">
          <DataField>UnitsInStock</DataField>
        </Field>
        <Field Name="UnitsOnOrder">
          <DataField>UnitsOnOrder</DataField>
        </Field>
        <Field Name="ReorderLevel">
          <DataField>ReorderLevel</DataField>
        </Field>
        <Field Name="Discontinued">
          <DataField>Discontinued</DataField>
        </Field>
      </Fields>
      <Query>
        <DataSourceName>DataSource1</DataSourceName>
        <CommandText>Select * From Products</CommandText>
      </Query>
    </DataSet>
  </DataSets>
  <Body>
    <ReportItems>
      <Textbox Name="BodyTextbox">
        <CanGrow>true</CanGrow>
        <Value>Body</Value>
        <Top>0.33333in</Top>
        <Left>0.3125in</Left>
        <Height>1in</Height>
        <Width>2.5in</Width>
        <Style>
          <BorderColor>
            <Default>Black</Default>
          </BorderColor>
          <BorderStyle>
            <Default>Solid</Default>
          </BorderStyle>
          <BorderWidth>
            <Default>1pt</Default>
          </BorderWidth>
          <BackgroundColor>LightSteelBlue</BackgroundColor>
        </Style>
      </Textbox>
    </ReportItems>
    <Height>2.5in</Height>
    <Style>
      <BackgroundColor>Khaki</BackgroundColor>
    </Style>
  </Body>
  <PageHeader>
    <Height>0.5in</Height>
    <PrintOnFirstPage>true</PrintOnFirstPage>
    <PrintOnLastPage>true</PrintOnLastPage>
```

```
 <ReportItems>
   <Textbox Name="HeaderTextbox">
     <Value>Header</Value>
     <Top>0.14583in</Top>
     <Left>0.3125in</Left>
     <Height>0.19792in</Height>
     <Width>5in</Width>
     <Style>
       <BorderColor>
         <Default>DarkBlue</Default>
       </BorderColor>
       <BorderStyle>
         <Default>Solid</Default>
       </BorderStyle>
       <BorderWidth>
         <Default>1pt</Default>
       </BorderWidth>
        <BackgroundColor>Gray</BackgroundColor>
        <PaddingLeft>2pt</PaddingLeft>
        <PaddingRight>2pt</PaddingRight>
        <PaddingTop>2pt</PaddingTop>
        <PaddingBottom>2pt</PaddingBottom>
      </Style>
    </Textbox>
  </ReportItems>
  <Style>
    <BackgroundColor>Silver</BackgroundColor>
  </Style>
</PageHeader>
<PageFooter>
  <Height>0.75in</Height>
  <PrintOnFirstPage>true</PrintOnFirstPage>
  <PrintOnLastPage>true</PrintOnLastPage>
  <ReportItems>
    <Textbox Name="FooterTextbox">
      <CanGrow>true</CanGrow>
      <Value>Footer</Value>
      <Top>0.25in</Top>
      <Left>0.3125in</Left>
      <Height>0.17708in</Height>
      <Width>5in</Width>
      <Style>
        <BorderColor>
          <Default>DarkBlue</Default>
        </BorderColor>
        <BorderStyle>
          <Default>Solid</Default>
        </BorderStyle>
        <BorderWidth>
          <Default>1pt</Default>
        </BorderWidth>
        <BackgroundColor>LightGray</BackgroundColor>
      </Style>
    </Textbox>
  </ReportItems>
  <Style>
    <BackgroundColor>Silver</BackgroundColor>
  </Style>
</PageFooter>
<PageWidth>8.5in</PageWidth>
<PageHeight>6in</PageHeight>
<InteractiveWidth>8.5in</InteractiveWidth>
<LeftMargin>0.5in</LeftMargin>
<RightMargin>0.5in</RightMargin>
<TopMargin>0.25in</TopMargin>
<BottomMargin>0.25in</BottomMargin>
<Width>6.5in</Width>
<Language>=User!Language</Language>
<ReportParameters>
```

```
    <ReportParameter Name="Product">
      <DataType>String</DataType>
      <DefaultValue>
        <DataSetReference>
          <DataSetName>DataSet1</DataSetName>
          <ValueField>ProductID</ValueField>
        </DataSetReference>
      </DefaultValue>
      <Prompt>Product</Prompt>
      <ValidValues>
        <DataSetReference>
          <DataSetName>DataSet1</DataSetName>
          <ValueField>ProductID</ValueField>
          <LabelField>ProductName</LabelField>
        </DataSetReference>
      </ValidValues>
      <MultiValue>true</MultiValue>
    </ReportParameter>
    <ReportParameter Name="Quantity">
      <DataType>Integer</DataType>
      <DefaultValue>
        <Values>
          <Value>0</Value>
        </Values>
      </DefaultValue>
      <Prompt>Quantity</Prompt>
      <ValidValues>
        <ParameterValues>
          <ParameterValue>
            <Value>0</Value>
            <Label>0</Label>
          </ParameterValue>
          <ParameterValue>
            <Value>10</Value>
            <Label>10</Label>
          </ParameterValue>
          <ParameterValue>
            <Value>50</Value>
            <Label>50</Label>
          </ParameterValue>
        </ParameterValues>
      </ValidValues>
    </ReportParameter>
  </ReportParameters>
</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.

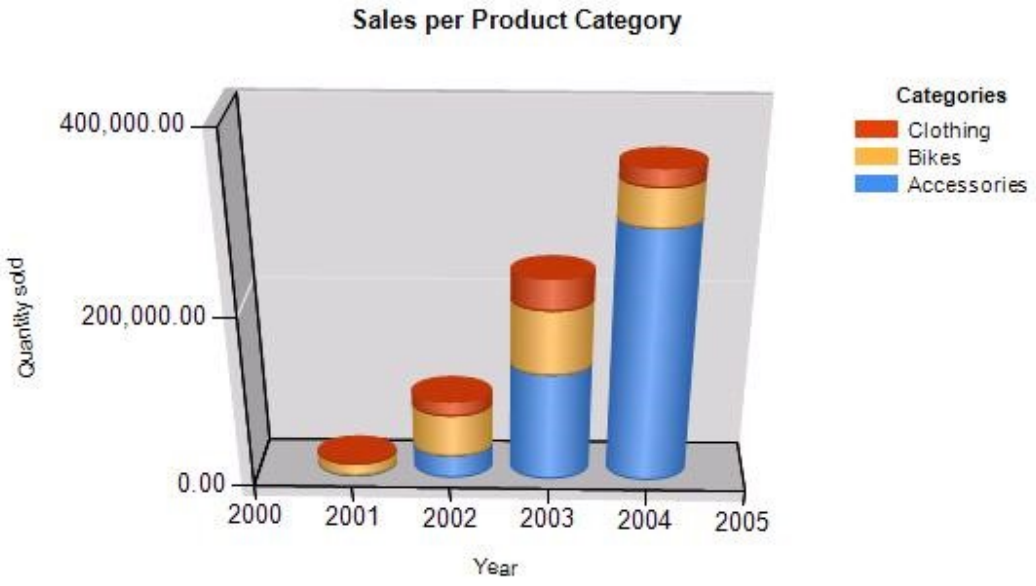

Figure 12: Chart1 example

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.

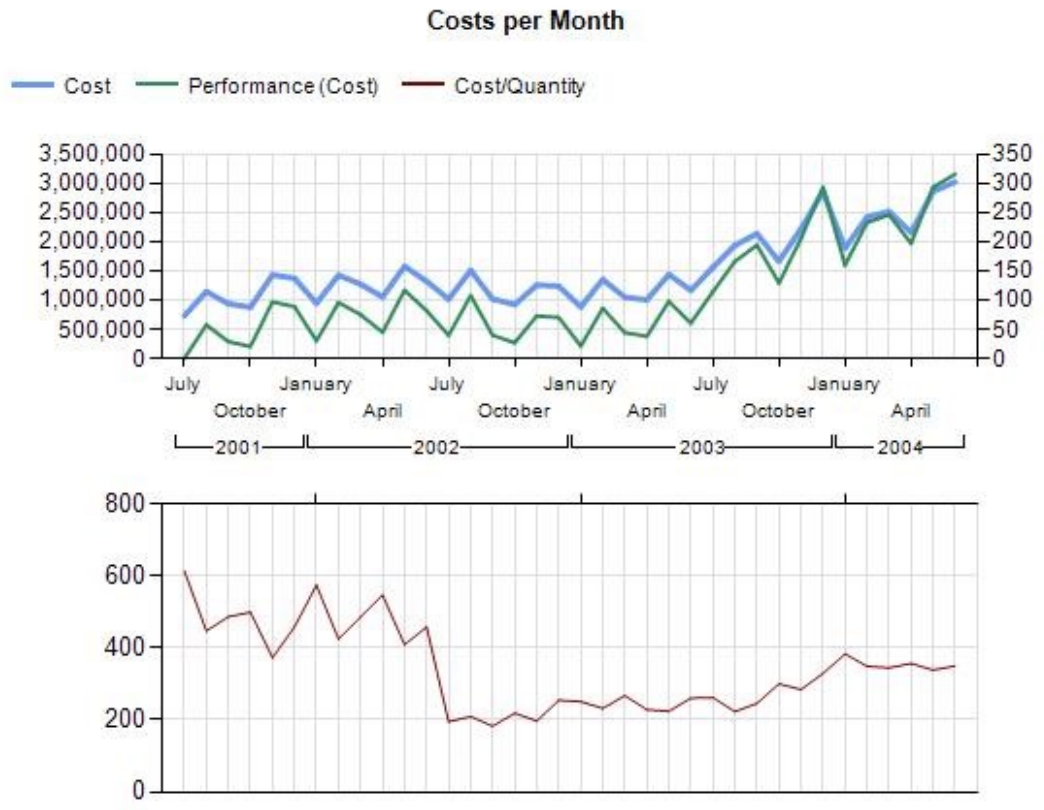

Figure 13: Chart2 example

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) &gt; 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               <Filters>
111                 <Filter>
112                   <FilterExpression>=Fields!ProductCategoryName.Value</FilterExpression>
113                   <Operator>NotEqual</Operator>
114                   <FilterValues>
115                     <FilterValue>Components</FilterValue>
116                   </FilterValues>
```

```
                </Filter>
              </Filters>
            </Group>
            <Label>=Fields!ProductCategoryName.Value</Label>
          </ChartMember>
        </ChartMembers>
      </ChartSeriesHierarchy>
      <ChartData>
        <ChartSeriesCollection>
          <ChartSeries Name="Quantity">
            <ChartDataPoints>
              <ChartDataPoint>
                <ChartDataPointValues>
                  <X>=Fields!Year.Value</X>
                  <Y>=Sum(Fields!Quantity.Value)</Y>
                </ChartDataPointValues>
                <ChartDataLabel>
                  <Style />
                </ChartDataLabel>
                <Style />
                <ChartMarker>
                  <Style />
                </ChartMarker>
                <DataElementOutput>Output</DataElementOutput>
              </ChartDataPoint>
            </ChartDataPoints>
            <Subtype>Stacked</Subtype>
            <Style />
            <ChartEmptyPoints>
              <Style />
              <ChartMarker>
                <Style />
              </ChartMarker>
              <ChartDataLabel>
                <Style />
              </ChartDataLabel>
            </ChartEmptyPoints>
            <CustomProperties>
              <CustomProperty>
                <Name>DrawingStyle</Name>
                <Value>Cylinder</Value>
              </CustomProperty>
            </CustomProperties>
            <ValueAxisName>Primary</ValueAxisName>
            <CategoryAxisName>Primary</CategoryAxisName>
            <ChartSmartLabel>
              <CalloutLineColor>Black</CalloutLineColor>
              <MinMovingDistance>0pt</MinMovingDistance>
            </ChartSmartLabel>
          </ChartSeries>
        </ChartSeriesCollection>
      </ChartData>
      <ChartAreas>
        <ChartArea Name="Default">
          <ChartCategoryAxes>
            <ChartAxis Name="Primary">
              <Style>
                <FontSize>8pt</FontSize>
              </Style>
              <ChartAxisTitle>
                <Caption>Year</Caption>
                <Style>
                  <FontSize>8pt</FontSize>
                </Style>
              </ChartAxisTitle>
              <ChartMajorGridLines>
                <Enabled>False</Enabled>
                <Style>
                  <Border>
```

```
                    <Color>Gainsboro</Color>
                  </Border>
                </Style>
              </ChartMajorGridLines>
              <ChartMinorGridLines>
                <Style>
                  <Border>
                    <Color>Gainsboro</Color>
                    <Style>Dotted</Style>
                  </Border>
                </Style>
              </ChartMinorGridLines>
              <ChartMinorTickMarks>
                <Length>0.5</Length>
              </ChartMinorTickMarks>

              <CrossAt>NaN</CrossAt>
              <Minimum>NaN</Minimum>
              <Maximum>NaN</Maximum>
            </ChartAxis>
            <ChartAxis Name="Secondary">
              <Style>
                <FontSize>8pt</FontSize>
              </Style>
              <ChartAxisTitle>
                <Caption>Axis Title</Caption>
                <Style>
                  <FontSize>8pt</FontSize>
                </Style>
              </ChartAxisTitle>
              <ChartMajorGridLines>
                <Enabled>False</Enabled>
                <Style>
                  <Border>
                    <Color>Gainsboro</Color>
                  </Border>
                </Style>
              </ChartMajorGridLines>
              <ChartMinorGridLines>
                <Style>
                  <Border>
                    <Color>Gainsboro</Color>
                    <Style>Dotted</Style>
                  </Border>
                </Style>
              </ChartMinorGridLines>
              <ChartMinorTickMarks>
                <Length>0.5</Length>
              </ChartMinorTickMarks>
              <CrossAt>NaN</CrossAt>
              <Location>Opposite</Location>
              <Minimum>NaN</Minimum>
              <Maximum>NaN</Maximum>
            </ChartAxis>
          </ChartCategoryAxes>
          <ChartValueAxes>
            <ChartAxis Name="Primary">
              <Style>
                <FontSize>8pt</FontSize>
                <Format>#,0.00;(#,0.00)</Format>
              </Style>
              <ChartAxisTitle>
                <Caption>Quantity sold</Caption>
                <Style>
                  <FontSize>8pt</FontSize>
                </Style>
              </ChartAxisTitle>
              <ChartMajorGridLines>
                <Style>
```

```
                  <Border>
                    <Color>Gainsboro</Color>
                  </Border>
                </Style>
              </ChartMajorGridLines>
              <ChartMinorGridLines>
                <Style>
                  <Border>
                    <Color>Gainsboro</Color>
                    <Style>Dotted</Style>
                  </Border>
                </Style>
              </ChartMinorGridLines>
              <ChartMinorTickMarks>
                <Length>0.5</Length>
              </ChartMinorTickMarks>
              <CrossAt>NaN</CrossAt>
              <Minimum>NaN</Minimum>
              <Maximum>NaN</Maximum>
            </ChartAxis>
            <ChartAxis Name="Secondary">
              <Style>
                <FontSize>8pt</FontSize>
              </Style>
              <ChartAxisTitle>
                <Caption>Axis Title</Caption>
                <Style>
                  <FontSize>8pt</FontSize>
                </Style>
              </ChartAxisTitle>
              <ChartMajorGridLines>
                <Style>
                  <Border>
                    <Color>Gainsboro</Color>
                  </Border>
                </Style>
              </ChartMajorGridLines>
              <ChartMinorGridLines>
                <Style>
                  <Border>
                    <Color>Gainsboro</Color>
                    <Style>Dotted</Style>
                  </Border>
                </Style>
              </ChartMinorGridLines>
              <ChartMinorTickMarks>
                <Length>0.5</Length>
              </ChartMinorTickMarks>
              <CrossAt>NaN</CrossAt>
              <Location>Opposite</Location>
              <Minimum>NaN</Minimum>
              <Maximum>NaN</Maximum>
            </ChartAxis>
          </ChartValueAxes>
          <ChartThreeDProperties>
            <Enabled>true</Enabled>
            <ProjectionMode>Perspective</ProjectionMode>
            <Perspective>9</Perspective>
            <Rotation>2</Rotation>
          </ChartThreeDProperties>
          <Style>
            <BackgroundGradientType>None</BackgroundGradientType>
          </Style>
        </ChartArea>
      </ChartAreas>
      <ChartLegends>
        <ChartLegend Name="Default">
          <Style>
            <BackgroundGradientType>None</BackgroundGradientType>
```

```
            <FontSize>8pt</FontSize>
          </Style>
          <DockToChartArea>Default</DockToChartArea>
          <DockOutsideChartArea>true</DockOutsideChartArea>
          <ChartLegendTitle>
            <Caption>Categories</Caption>
            <Style>
              <FontSize>8pt</FontSize>
              <FontWeight>Bold</FontWeight>
              <TextAlign>Center</TextAlign>
            </Style>
          </ChartLegendTitle>
          <HeaderSeparatorColor>Black</HeaderSeparatorColor>
          <ColumnSeparatorColor>Black</ColumnSeparatorColor>
        </ChartLegend>
      </ChartLegends>
      <ChartTitles>
        <ChartTitle Name="Default">
          <Caption>Sales per Product Category</Caption>
          <Style>
            <BackgroundGradientType>None</BackgroundGradientType>
            <FontWeight>Bold</FontWeight>
            <TextAlign>General</TextAlign>
            <VerticalAlign>Top</VerticalAlign>
          </Style>
        </ChartTitle>
      </ChartTitles>
      <Palette>BrightPastel</Palette>
      <ChartBorderSkin>
        <Style>
          <BackgroundColor>Gray</BackgroundColor>
          <BackgroundGradientType>None</BackgroundGradientType>
          <Color>White</Color>
        </Style>
      </ChartBorderSkin>
      <ChartNoDataMessage Name="NoDataMessage">
        <Caption>No Data Available</Caption>
        <Style>
          <BackgroundGradientType>None</BackgroundGradientType>
          <TextAlign>General</TextAlign>
          <VerticalAlign>Top</VerticalAlign>
        </Style>
      </ChartNoDataMessage>
      <DataSetName>SalesPerProduct</DataSetName>
      <Top>0.25708in</Top>
      <Left>0.21542in</Left>
      <Height>3.53125in</Height>
      <Width>6.08333in</Width>
      <Style>
        <Border>
          <Color>LightGrey</Color>
          <Style>Solid</Style>
        </Border>
        <BackgroundColor>White</BackgroundColor>
        <BackgroundGradientType>None</BackgroundGradientType>
      </Style>
    </Chart>
    <Chart Name="Chart2">
      <ChartCategoryHierarchy>
        <ChartMembers>
          <ChartMember>
            <Group Name="Chart2_CategoryGroup1">
              <GroupExpressions>
                <GroupExpression>=Fields!Year.Value</GroupExpression>
              </GroupExpressions>
            </Group>
            <ChartMembers>
              <ChartMember>
                <Group Name="Chart2_CategoryGroup2">
```

```
                  <GroupExpressions>
                    <GroupExpression>=Fields!Month.Value</GroupExpression>
                  </GroupExpressions>
                </Group>
                <Label>=MonthName(Fields!Month.Value)</Label>
              </ChartMember>
            </ChartMembers>
            <Label>=Fields!Year.Value</Label>
          </ChartMember>
        </ChartMembers>
      </ChartCategoryHierarchy>
      <ChartSeriesHierarchy>
        <ChartMembers>
          <ChartMember>
            <Label>Cost</Label>
          </ChartMember>
          <ChartMember>
            <Label>Cost</Label>
          </ChartMember>
        </ChartMembers>
      </ChartSeriesHierarchy>
      <ChartData>
        <ChartSeriesCollection>
          <ChartSeries Name="Cost">
            <ChartDataPoints>
              <ChartDataPoint>
                <ChartDataPointValues>
                  <Y>=Sum(Fields!Cost.Value)</Y>
                </ChartDataPointValues>
                <ChartDataLabel>
                  <Style />
                </ChartDataLabel>
                <Style>
                  <Border>
                    <Width>2pt</Width>
                  </Border>
                </Style>
                <ChartMarker>
                  <Style />
                </ChartMarker>
                <DataElementOutput>Output</DataElementOutput>
              </ChartDataPoint>
            </ChartDataPoints>
            <Type>Line</Type>
            <Style />
            <ChartEmptyPoints>
              <Style />
              <ChartMarker>
                <Style />
              </ChartMarker>
              <ChartDataLabel>
                <Style />
              </ChartDataLabel>
            </ChartEmptyPoints>
            <ValueAxisName>Primary</ValueAxisName>
            <CategoryAxisName>Primary</CategoryAxisName>
            <ChartSmartLabel>
              <CalloutLineColor>Black</CalloutLineColor>
              <MinMovingDistance>0pt</MinMovingDistance>
            </ChartSmartLabel>
          </ChartSeries>
          <ChartSeries Name="Cost1">
            <ChartDataPoints>
              <ChartDataPoint>
                <ChartDataPointValues>
                  <Y>=Sum(Fields!Cost.Value) / Sum(Fields!Quantity.Value)</Y>
                </ChartDataPointValues>
                <ChartDataLabel>
                  <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>
```

```
              <Location>Opposite</Location>
              <Minimum>NaN</Minimum>
              <Maximum>NaN</Maximum>
            </ChartAxis>
          </ChartValueAxes>
          <Style>
            <BackgroundGradientType>None</BackgroundGradientType>
          </Style>
        </ChartArea>
        <ChartArea Name="ChartArea2">
          <ChartCategoryAxes>
            <ChartAxis Name="Primary">
              <Style>
                <FontSize>8pt</FontSize>
              </Style>
              <ChartAxisTitle>
                <Caption />
                <Style>
                  <FontSize>8pt</FontSize>
                </Style>
              </ChartAxisTitle>
              <ChartMajorGridLines>
                <Enabled>True</Enabled>
                <Style>
                  <Border>
                    <Color>Gainsboro</Color>
                  </Border>
                </Style>
                <Interval>1</Interval>
              </ChartMajorGridLines>
              <ChartMinorGridLines>
                <Style>
                  <Border>
                    <Color>Gainsboro</Color>
                    <Style>Dotted</Style>
                  </Border>
                </Style>
              </ChartMinorGridLines>
              <ChartMajorTickMarks>
                <Enabled>False</Enabled>
              </ChartMajorTickMarks>
              <ChartMinorTickMarks>
                <Length>0.5</Length>
              </ChartMinorTickMarks>
              <CrossAt>NaN</CrossAt>
              <Minimum>NaN</Minimum>
              <Maximum>NaN</Maximum>
              <HideLabels>true</HideLabels>
            </ChartAxis>
            <ChartAxis Name="Secondary">
              <Visible>True</Visible>
              <Style>
                <FontSize>8pt</FontSize>
              </Style>
              <ChartAxisTitle>
                <Caption />
                <Style>
                  <FontSize>8pt</FontSize>
                </Style>
              </ChartAxisTitle>
              <ChartMajorGridLines>
                <Enabled>False</Enabled>
                <Style>
                  <Border>
                    <Color>Gainsboro</Color>
                  </Border>
                </Style>
              </ChartMajorGridLines>
              <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>
```

```
                    <Color>Gainsboro</Color>
                  </Border>
                </Style>
              </ChartMajorGridLines>
              <ChartMinorGridLines>
                <Style>
                  <Border>
                    <Color>Gainsboro</Color>
                    <Style>Dotted</Style>
                  </Border>
                </Style>
              </ChartMinorGridLines>
              <ChartMinorTickMarks>
                <Length>0.5</Length>
              </ChartMinorTickMarks>
              <CrossAt>NaN</CrossAt>
              <Location>Opposite</Location>
              <Minimum>NaN</Minimum>
              <Maximum>NaN</Maximum>
            </ChartAxis>
          </ChartValueAxes>
          <Style>
            <BackgroundGradientType>None</BackgroundGradientType>
          </Style>
          <AlignOrientation>Vertical</AlignOrientation>
          <ChartAlignType>
            <InnerPlotPosition>true</InnerPlotPosition>
          </ChartAlignType>
          <AlignWithChartArea>Default</AlignWithChartArea>
        </ChartArea>
      </ChartAreas>
      <ChartLegends>
        <ChartLegend Name="Default">
          <Style>
            <BackgroundGradientType>None</BackgroundGradientType>
            <FontSize>8pt</FontSize>
          </Style>
          <Position>TopLeft</Position>
          <DockOutsideChartArea>true</DockOutsideChartArea>
          <ChartLegendTitle>
            <Caption />
            <Style>
              <FontSize>8pt</FontSize>
              <FontWeight>Bold</FontWeight>
              <TextAlign>Center</TextAlign>
            </Style>
          </ChartLegendTitle>
          <HeaderSeparatorColor>Black</HeaderSeparatorColor>
          <ColumnSeparatorColor>Black</ColumnSeparatorColor>
        </ChartLegend>
      </ChartLegends>
      <ChartTitles>
        <ChartTitle Name="Default">
          <Caption>Costs per Month</Caption>
          <Style>
            <BackgroundGradientType>None</BackgroundGradientType>
            <FontWeight>Bold</FontWeight>
            <TextAlign>General</TextAlign>
            <VerticalAlign>Top</VerticalAlign>
          </Style>
        </ChartTitle>
      </ChartTitles>
      <Palette>BrightPastel</Palette>
      <ChartBorderSkin>
        <Style>
          <BackgroundColor>Gray</BackgroundColor>
          <BackgroundGradientType>None</BackgroundGradientType>
          <Color>White</Color>
        </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**.

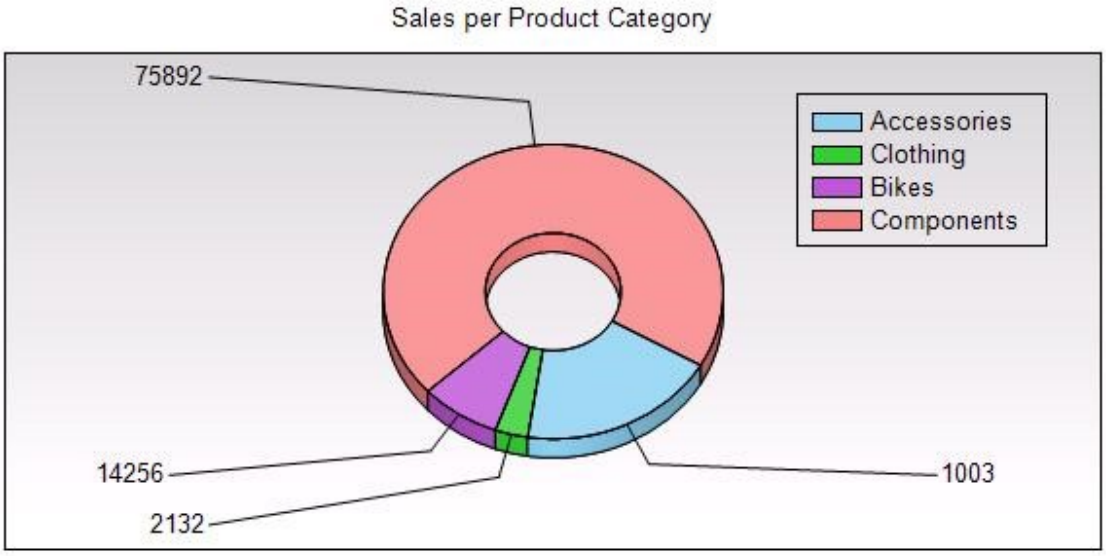

Figure 14: Doughnut chart example

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.

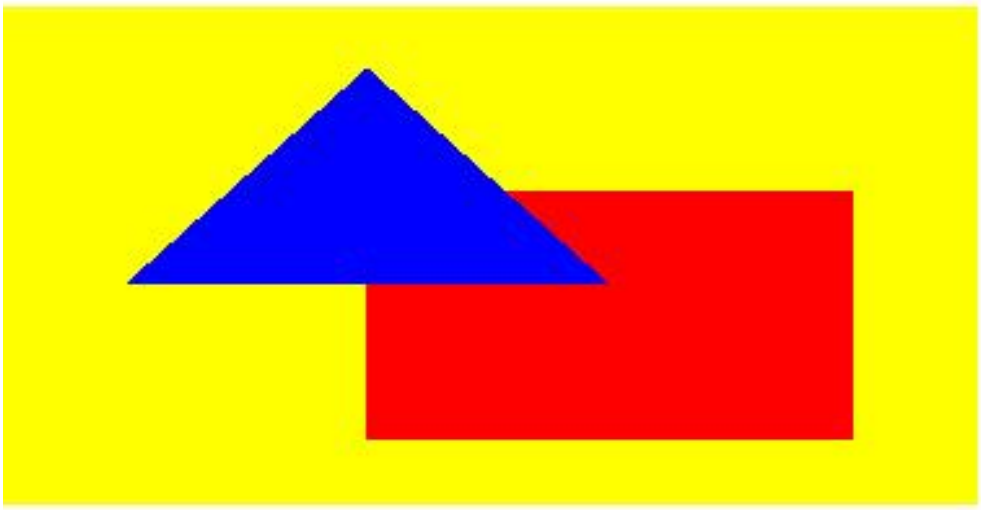

Figure 15: A data source is defined as embedded XML.

```
<DataSources>
  <DataSource Name="PolygonsData">
    <ConnectionProperties>
      <DataProvider>XML</DataProvider>
      <ConnectString />
    </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 ("<" and ">") 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>
&lt;Query&gt;
  &lt;XmlData&gt;
    &lt;Rows&gt;
      &lt;Row&gt;
        &lt;Object&gt;1&lt;/Object&gt;&lt;Color&gt;Red&lt;/Color&gt;
        &lt;Point&gt;1&lt;/Point&gt;&lt;X&gt;75&lt;/X&gt;&lt;Y&gt;75&lt;/Y&gt;
      &lt;/Row&gt;
      &lt;Row&gt;
        &lt;Object&gt;1&lt;/Object&gt;&lt;Color&gt;Red&lt;/Color&gt;
        &lt;Point&gt;2&lt;/Point&gt;&lt;X&gt;75&lt;/X&gt;&lt;Y&gt;175&lt;/Y&gt;
      &lt;/Row&gt;
      &lt;Row&gt;
        &lt;Object&gt;1&lt;/Object&gt;&lt;Color&gt;Red&lt;/Color&gt;
        &lt;Point&gt;3&lt;/Point&gt;&lt;X&gt;175&lt;/X&gt;&lt;Y&gt;175&lt;/Y&gt;
      &lt;/Row&gt;
      &lt;Row&gt;
        &lt;Object&gt;1&lt;/Object&gt;&lt;Color&gt;Red&lt;/Color&gt;
        &lt;Point&gt;4&lt;/Point&gt;&lt;X&gt;175&lt;/X&gt;&lt;Y&gt;75&lt;/Y&gt;
      &lt;/Row&gt;
      &lt;Row&gt;
        &lt;Object&gt;2&lt;/Object&gt;&lt;Color&gt;Blue&lt;/Color&gt;
        &lt;Point&gt;1&lt;/Point&gt;&lt;X&gt;75&lt;/X&gt;&lt;Y&gt;25&lt;/Y&gt;
      &lt;/Row&gt;
      &lt;Row&gt;
        &lt;Object&gt;2&lt;/Object&gt;&lt;Color&gt;Blue&lt;/Color&gt;
        &lt;Point&gt;2&lt;/Point&gt;&lt;X&gt;25&lt;/X&gt;&lt;Y&gt;112&lt;/Y&gt;
      &lt;/Row&gt;
      &lt;Row&gt;
        &lt;Object&gt;2&lt;/Object&gt;&lt;Color&gt;Blue&lt;/Color&gt;
        &lt;Point&gt;3&lt;/Point&gt;&lt;X&gt;125&lt;/X&gt;&lt;Y&gt;112&lt;/Y&gt;
      &lt;/Row&gt;
    &lt;/Rows&gt;
  &lt;/XmlData&gt;
&lt;ElementPath&gt;Rows/Row{Object(Integer),Color(String),Point(Integer),X(Integer),Y(Integer
)}&lt;/ElementPath&gt;
&lt;/Query&gt;
      </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	**Abingdon** England OX14 4SE	United Kingdom
2255 254th Avenue Se	**Albany** Oregon 97321	United States
Heritage Mall	**Albany** Oregon 97321	United States
Vista Marketplace	**Alhambra** California 91801	United States
Viejas Outlet Center	**Alpine** California 91901	United States
Lincoln Square	**Arlington** Texas 76010	United States
99700 Bell Road	**Auburn** California 95603	United States
32605 West 252 Mile Road, Suite 250	**Aurora** 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>
```

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>
        <Style>
          <Border>
            <Color>LightGrey</Color>
            <Style>Solid</Style>
          </Border>
          <BackgroundColor>Aqua</BackgroundColor>
        </Style>
      </Textbox>
    </CellContents>
  </TablixCell>
  <TablixCell>
    <CellContents>
      <Textbox Name="Heading2">
        <Paragraphs>
          <Paragraph>
            <TextRuns>
              <TextRun>
                <Value>City</Value>
              </TextRun>
            </TextRuns>
          </Paragraph>
        </Paragraphs>
        <Style>
          <Border>
            <Color>LightGrey</Color>
            <Style>Solid</Style>
          </Border>
          <BackgroundColor>Aqua</BackgroundColor>
        </Style>
      </Textbox>
    </CellContents>
  </TablixCell>
  <TablixCell>
    <CellContents>
      <Textbox Name="Heading3">
        <KeepTogether>true</KeepTogether>
        <Paragraphs>
          <Paragraph>
            <TextRuns>
              <TextRun>
                <Value>Country</Value>
              </TextRun>
            </TextRuns>
          </Paragraph>
        </Paragraphs>
        <Style>
          <Border>
            <Color>LightGrey</Color>
            <Style>Solid</Style>
          </Border>
          <BackgroundColor>Aqua</BackgroundColor>
        </Style>
      </Textbox>
    </CellContents>
```

```
    </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>
                  <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	**First order received**	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
	Total	211.8400	94.0570	755.9200	335.6287	21
Helmets	**First order received**	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
	Total	244.9300	91.6041	349.9000	130.8630	111
Hydration Packs	**First order received**	Jun 01, 2004		Jun 01, 2004		
	Hydration Pack - 70 oz.	109.9800	41.1326	219.9600	82.2652	45
	Total	109.9800	41.1326	219.9600	82.2652	45
Jerseys	**First order received**	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
	Long-Sleeve Logo Jersey, XL	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>
    <TablixHeader>
```

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>
                <Style>
                  <TextAlign>Right</TextAlign>
                </Style>
              </Paragraph>
            </Paragraphs>
            <Style>
              <Border>
                <Color>LightGrey</Color>
                <Style>Solid</Style>
              </Border>
              <BackgroundColor>Aqua</BackgroundColor>
            </Style>
          </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>
            <Style>
              <Border>
                <Color>LightGrey</Color>
                <Style>Solid</Style>
              </Border>
              <BackgroundColor>Aqua</BackgroundColor>
            </Style>
          </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>
```

```
<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>
```

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 2.22 and 2.36.1:

"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 2.22 and 2.37.1:

"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>
```

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>
                  </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>
            <Style>
              <Border>
                <Color>LightGrey</Color>
                <Style>Solid</Style>
              </Border>
            </Style>
          </Textbox>
        </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>
```

```
                  <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.

```
<DataSources>
 <DataSource Name="DataSource1">
   <ConnectionProperties>
     <DataProvider>SQL</DataProvider>
     <ConnectString>data source=dataServer; initial
                    catalog=northwind;</ConnectString>
     <IntegratedSecuity>true</IntegratedSecuity>
   </ConnectionProperties>
 </DataSource>
</DataSources>
<DataSets>
 <DataSet Name="DataSet1">
   <Fields>
     <Field Name="ProductID">
       <DataField>ProductID</DataField>
     </Field>
     <Field Name="ProductName">
       <DataField>ProductName</DataField>
     </Field>
     <Field Name="SupplierID">
       <DataField>SupplierID</DataField>
     </Field>
     <Field Name="CategoryID">
       <DataField>CategoryID</DataField>
     </Field>
     <Field Name="QuantityPerUnit">
       <DataField>QuantityPerUnit</DataField>
     </Field>
     <Field Name="UnitPrice">
       <DataField>UnitPrice</DataField>
     </Field>
     <Field Name="UnitsInStock">
       <DataField>UnitsInStock</DataField>
     </Field>
     <Field Name="UnitsOnOrder">
       <DataField>UnitsOnOrder</DataField>
     </Field>
     <Field Name="ReorderLevel">
       <DataField>ReorderLevel</DataField>
     </Field>
     <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 &gt;= @UnitsInStockMin
                  AND UnitsInStock &lt;= @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.

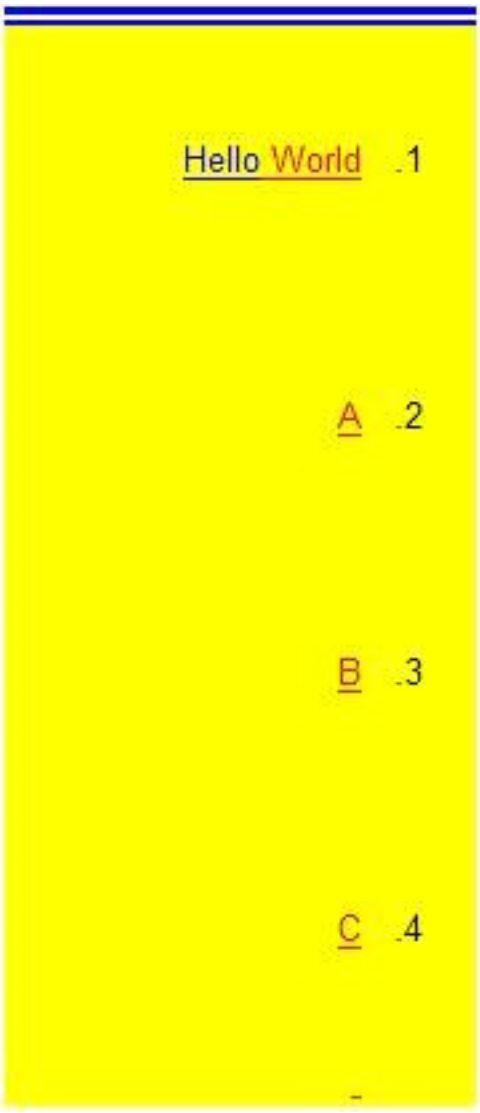

Figure 20: Text box example

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="GaugePanel3">
  <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>
```

```
            <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.

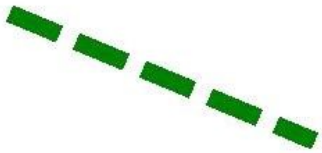

Figure 22: Line example

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.

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.

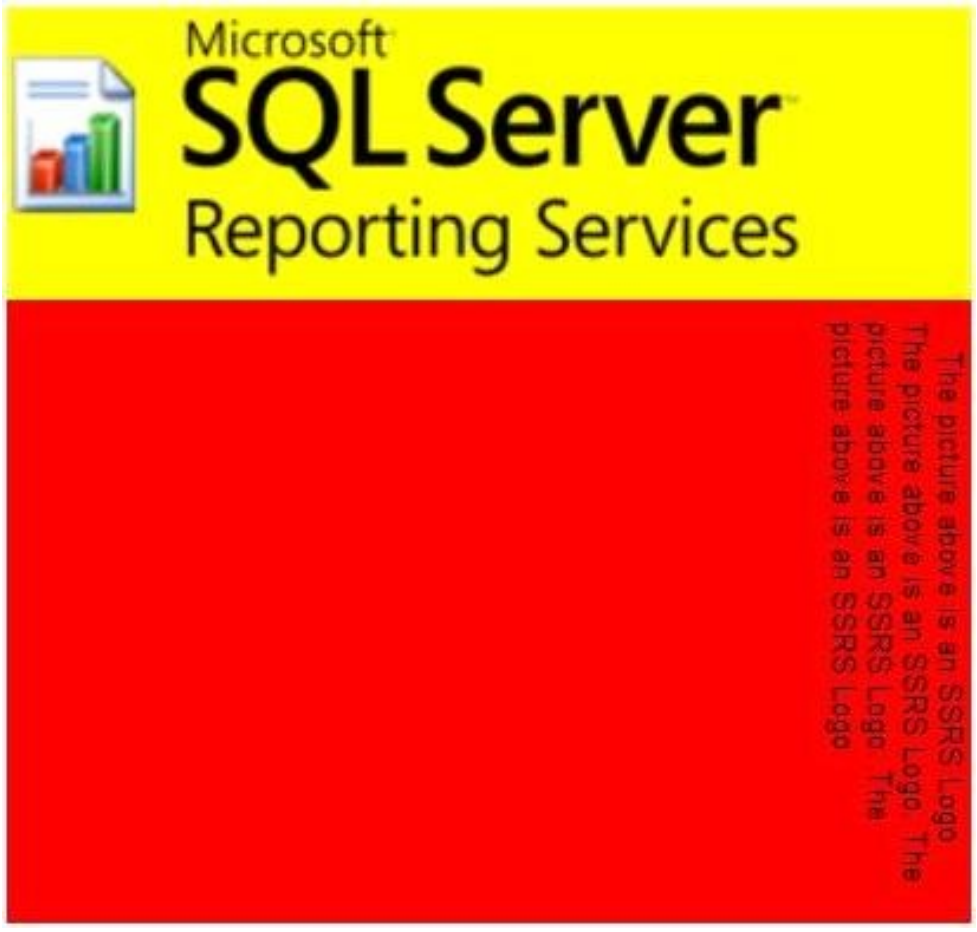

Figure 24: Rectangle with an image and a text box

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>
```

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>
```

```
          <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.

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>
        <ConnectString>Data Source=.\map7;Initial Catalog=Election</ConnectString>
        <IntegratedSecurity>true</IntegratedSecurity>
      </ConnectionProperties>
      <rd:DataSourceID>b1cba34d-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>
      </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>
```

```
      <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 />
        <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>
<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>
        <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>
      <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>
        <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
	Subtotal		3827950	2396540
	Germany	Rachel B Valdez	2241204	1307950
	Subtotal		2241204	1307950
	United Kingdom	Jae B Pak	5015682	1635823
	Subtotal		5015682	1635823
Total			11084837	5340313
North America	Canada	José Edvaldo Saraiva	3189356	2038235
		Garrett R Vargas	1764939	1620277
	Subtotal		4954295	3658512
	Central	Jillian Carson	3857164	1997186
	Subtotal		3857164	1997186
	Northeast	Michael G Blythe	4557045	1750406
	Subtotal		4557045	1750406
	Northwest	Pamela O Ansman-Wolfe	0	1927059
		David R Campbell	3587378	1371635
		Tete A Mensa-Annan	1931620	0
	Subtotal		5518999	3298694
	Southeast	Tsvi Michael Reiter	2811013	1849641
	Subtotal		2811013	1849641
	Southwest	Shu K Ito	3018725	2073506
		Linda C Mitchell	5200475	1439156
	Subtotal		8219201	3512662
Total			29917716	16067102
Pacific	Australia	Lynn N Tsoflias	1758386	2278549
	Subtotal		1758386	2278549
Total			1758386	2278549
Grand Total			42760939	23685964

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>
```

```
                <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>
```

```
                    <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>
        </TableCell>
      </TableCells>
      <Height>0.21in</Height>
    </TableRow>
  </TableRows>
```

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 Ansman-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>
```

```
          <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>
        <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>
        </TableCell>
      </TableCells>
      <Height>0.25in</Height>
    </TableRow>
  </TableRows>
</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	5.1
RDL XML Schema for Version 2005/01	xsd	5.2
RDL XML Schema for Version 2008/01	xsd	5.3
RDL XML Schema for Version 2010/01	xsd	5.4
RDL XML Schema for Version 2011/01*	xsd	5.5
RDL XML Schema for Version 2012/01*	xsd	5.6
RDL XML Schema for Version 2013/01*	xsd	5.7
RDL XML Schema for Version 2016/01	xsd	5.8

*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: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: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: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: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: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: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: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="ChartDataType" 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: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="ChartDataType">
    <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: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: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: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: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: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: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: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: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: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: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: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: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" 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="ChartDataType" 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>
<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: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: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="ChartDataType">
    <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:any namespace="##other" processContents="skip"/>
    </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:any namespace="##other" processContents="skip"/>
    </xsd:choice>
    <xsd:anyAttribute 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: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: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: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: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="ValuesType">
  <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="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="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="StringWithDataTypeAttribute"
                 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="StringWithDataTypeAttribute" />
    <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: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="LocIDStringWithDataTypeAttribute"
                 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: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: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="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>
  <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: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="ChartDataType" 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: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: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: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: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: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="ChartDataType">
  <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="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: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: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: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: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="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>
  <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: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="StringWithDataTypeAttribute"
                   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="LocIDStringWithDataTypeAttribute">
    <xsd:simpleContent>
      <xsd:extension base="StringWithDataTypeAttribute">
        <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: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:anyAttribute namespace="##other" processContents="skip" />
  </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: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: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: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: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="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>
  <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="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: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: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 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: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>
<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: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="StringWithDataTypeAttribute" />
      <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: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="LocIDStringWithDataTypeAttribute" 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: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="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="ChartDataType" 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: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: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: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="ChartDataType">
   <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: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: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: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: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: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: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: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: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: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: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: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>

  <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: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: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-->
    <!--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: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: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: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: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: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 unqiue 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: rdl2010:ChartSeries  -->
  <xsd:element name ="Subtype" type="xsd:string" />

  <!-- Indicates how the report embeds the images.
      ParentElements: rdl2010:Image, rdl2010: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: rdl2010: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: rdl2010:TablixRowHierarchy, rdl2010:TablixColumnHierarchy,
rdl2010: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
targetNamespace="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 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="DefaultFontFamily" type="xsd:string" minOccurs="0"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2016/01/reportdefinition/defaultfontf
amily" />
            <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 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: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="StringWithDataTypeAttribute" />
            <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: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="LocIDStringWithDataTypeAttribute" 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: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="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="ChartDataType" 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: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: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: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="ChartDataType">
         <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: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: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: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: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: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: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: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: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: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: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: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>

  <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: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: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-->
        <!--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: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: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: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 **DateTime** 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 rsCRIStaticWithSubgroups

Applies to RDL 2003/10 and RDL 2005/01

rsCRIStaticWithSubgroups 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 2.340.7).

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 NaturalSortand 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

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 2019 Reporting Services supports all previous RDL versions.

RDL schema version	Release introduced
RDL 2003/10	Microsoft SQL Server 2000 Reporting Services
RDL 2005/01	Microsoft SQL Server 2005 Reporting Services
RDL 2008/01	Microsoft SQL Server 2008 Reporting Services
RDL 2010/01	Microsoft SQL Server 2008 R2 Reporting Services
RDL 2011/01	Microsoft SQL Server 2012 Reporting Services
RDL 2012/01	Microsoft SQL Server 2012 Service Pack 1 (SP1) Reporting Services
RDL 2013/01	SQL Server 2012 SP1 Reporting Services
RDL 2016/01	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 2.3.6: In the Microsoft implementation, the code is specified in a Visual Basic–compatible syntax [MSFT-VBNET].

<4> Section 2.6.1: The value of the Body.Columns element is less than or equal to 1000.

<5> Section 2.7.1: The value of the **Page.Columns** element is less than or equal to 1000.

<6> Section 2.17.3: The value of the **Paragraph.ListLevel** element is less than or equal to 9.

<7> Section 2.22.10: 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 2.22.18: 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 2.29.2: 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 2.29.4: 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 2.29.4: 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 2.33.7: 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 2.80.9: 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 2.80.9: 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 2.80.9: 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 2.80.10: 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 2.83.2: 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 2.83.2: 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 2.83.2: 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 2.83.2: 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 2.83.2: 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 2.87.24: 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 2.120.2: The value of the **ChartAxisScaleBreak.CollapsibleSpaceThreshold** element is greater than or equal to 10 and less than or equal to 90.

<24> Section 2.120.5: The value of the **ChartAxisScaleBreak.MaxNumberOfBreaks** element is less than or equal to 5.

<25> Section 2.120.6: The value of the **ChartAxisScaleBreak.Spacing** element is less than or equal to 10.

<26> Section 2.127.2: The value of the **ChartThreeDProperties.DepthRatio** element is less than or equal to 1000.

<27> Section 2.127.8: The value of the **ChartThreeDProperties.Rotation** element is greater than or equal to -180 and less than or equal to 180.

<28> Section 2.127.10: The value of the **ChartThreeDProperties.WallThickness** element is less than or equal to 30.

<29> Section 2.140.12: 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 2.140.12: 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 2.145.6: The value of the **ChartSmartLabel.CalloutLineWidth** element is greater than or equal to 0.24985pt and less than or equal to 20pt.

<32> Section 2.151.8: The value of the **ChartLegend.ColumnSpacing** element is less than or equal to 100.

<33> Section 2.162.17: 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 2.176.3: The value of the **CustomLabel.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

<35> Section 2.177.1: The value of the **TickMarkStyle.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

<36> Section 2.177.5: The value of the **TickMarkStyle.Length** element is less than or equal to 100.

<37> Section 2.177.10: The value of the **TickMarkStyle.Width** element is less than or equal to 100.

<38> Section 2.179.3: The value of the **GaugeTickMarks.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

<39> Section 2.179.7: The value of the **GaugeTickMarks.Length** element is less than or equal to 100.

<40> Section 2.179.12: The value of the **GaugeTickMarks.Width** element is less than or equal to 100.

<41> Section 2.181.4: The value of the **LinearPointer.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

<42> Section 2.181.7: The value of the **LinearPointer.MarkerLength** element is less than or equal to 100.

<43> Section 2.183.1: The value of the **Thermometer.BulbOffset** element is less than or equal to 100.

<44> Section 2.183.2: The value of the **Thermometer.BulbSize** element is less than or equal to 1000.

<45> Section 2.184.2: The value of the **ScaleLabels.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

<46> Section 2.185.1: The value of the **ScalePin.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

<47> Section 2.185.5: The value of the **ScalePin.Length** element is less than or equal to 100.

<48> Section 2.185.10: The value of the **ScalePin.Width** element is less than or equal to 100.

<49> Section 2.186.2: The value of the **PinLabel.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

<50> Section 2.188.4: The value of the **ScaleRange.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

<51> Section 2.188.6: The value of the **ScaleRange.EndWidth** element is less than or equal to 1000.

<52> Section 2.188.13: The value of the **ScaleRange.StartWidth** element is less than or equal to 1000.

<53> Section 2.199.4: The value of the **RadialPointer.DistanceFromScale** element is greater than or equal to -100 and less than or equal to 100.

<54> Section 2.199.7: The value of the **RadialPointer.MarkerLength** element is less than or equal to 100.

<55> Section 2.200.7: The value of the **PointerCap.Width** element is less than or equal to 1000.

<56> Section 2.215.2: 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 2.270.4: 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 2.277.2: 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 2.277.2: 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 2.283.4: 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 2.309.1: Microsoft implementations require this value to be greater than or equal to 1 and less than or equal to 8.

<62> Section 2.309.2: Microsoft implementations require this value to be greater than or equal to 1 and less than or equal to 10000.

<63> Section 2.310.1: Microsoft implementations require the number of consecutive empty rows to be less than or equal to 20.

<64> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Albanian_100 locale.

<65> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Amharic_100 locale.

<66> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Arabic_100 locale.

<67> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Armenian_100 locale.

<68> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Assamese_100 locale.

<69> Section 2.324.4: SQL Server 2000 Reporting Services does not support the Azeri_Cyrillic_90 locale.

<70> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Azeri_Cyrillic_100 locale.

<71> Section 2.324.4: SQL Server 2000 Reporting Services does not support the Azeri_Latin_90 locale.

<72> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Azeri_Latin_100 locale.

<73> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Bashkir_100 locale.

<74> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Bengali_100 locale.

<75> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Bosnian_Cyrillic_100 locale.

<76> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Bosnian_Latin_100 locale.

<77> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Breton_100 locale.

<78> Section 2.324.4: SQL Server 2000 Reporting Services does not support the Chinese_Hong_Kong_Stroke_90 locale.

<79> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese_Hong_Kong_Stroke_100 locale.

<80> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese_Macao_100 locale.

<81> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese_Macao_Stroke_100 locale.

<82> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese_PRC_100 locale.

<83> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese_PRC_Stroke_100 locale.

<84> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese_Simplified_Pinyin_100 locale.

<85> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese_Simplified_Stroke_Order_100 locale.

<86> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese_Taiwan_Bopomofo_100 locale.

<87> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese_Taiwan_Stroke_100 locale.

<88> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese_Traditional_Bopomofo_100 locale.

<89> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese_Traditional_Pinyin_100 locale.

<90> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese_Traditional_Stroke_Count_100 locale.

<91> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Chinese_Traditional_Stroke_Order_100 locale.

<92> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Corsican_100 locale.

<93> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Croatian_100 locale.

<94> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Cyrillic_General_100 locale.

<95> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Czech_100 locale.

<96> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Danish_Greenlandic_100 locale.

<97> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Dari_100 locale.

<98> Section 2.324.4: SQL Server 2000 Reporting Services does not support the Divehi_90 locale.

<99> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Divehi_100 locale.

<100> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Estonian_100 locale.

<101> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Finnish_Swedish_100 locale.

<102> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the French_100 locale.

<103> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Frisian_100 locale.

<104> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Georgian_Traditional_100 locale.

<105> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the German_PhoneBook_100 locale.

<106> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Greek_100 locale.

<107> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Hebrew_100 locale.

<108> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Hungarian_100 locale.

<109> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Hungarian_Technical_100 locale.

<110> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Icelandic_100 locale.

<111> Section 2.324.4: SQL Server 2000 Reporting Services does not support the Indic_General_90 locale.

<112> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Indic_General_100 locale.

<113> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Inuktitut_100 locale.

<114> Section 2.324.4: SQL Server 2000 Reporting Services does not support the Japanese_90 locale.

<115> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Japanese_100 locale.

<116> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Japanese_Bushu_Kakusu_100 locale.

<117> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Japanese_Radical_Stroke_100 locale.

<118> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Japanese_Unicode locale.

<119> Section 2.324.4: SQL Server 2000 Reporting Services does not support the Kazakh_90 locale.

<120> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Kazakh_100 locale.

<121> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Khmer_100 locale.

<122> Section 2.324.4: SQL Server 2000 Reporting Services does not support the Korean_90 locale.

<123> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Korean_100 locale.

<124> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Lao_100 locale.

<125> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Latin1_General_100 locale.

<126> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Latvian_100 locale.

<127> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Lithuanian_100 locale.

<128> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Lithuanian_Classic locale.

<129> Section 2.324.4: SQL Server 2000 Reporting Services does not support the Macedonian_FYROM_90 locale.

<130> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Macedonian_FYROM_100 locale.

<131> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Maltese_100 locale.

<132> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Maori_100 locale.

<133> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Mapudungan_100 locale.

<134> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Modern_Spanish_100 locale.

<135> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Mohawk_100 locale.

<136> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Mongolian_100 locale.

<137> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Nepali_100 locale.

<138> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Norwegian_100 locale.

<139> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Norwegian_Sami_100 locale.

<140> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Pashto_100 locale.

<141> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Persian_100 locale.

<142> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Polish_100 locale.

<143> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Romanian_100 locale.

<144> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Romansh_100 locale.

<145> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Sami_Norway_100 locale.

<146> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Sami_Sweden_Finland_100 locale.

<147> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Serbian_Cyrillic_100 locale.

<148> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Serbian_Latin_100 locale.

<149> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Slovak_100 locale.

<150> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Slovenian_100 locale.

<151> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Swedish_Finnish_Sami_100 locale.

<152> Section 2.324.4: SQL Server 2000 Reporting Services does not support the Syriac_90 locale.

<153> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Syriac_100 locale.

<154> Section 2.324.4: SQL Server 2000 Reporting Services does not support the Tatar_90 locale.

<155> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Tatar_100 locale.

<156> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Tamazight_100 locale.

<157> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Thai_100 locale.

<158> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Tibetan_PRC_100 locale.

<159> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Traditional_Spanish_100 locale.

<160> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Turkish_100 locale.

<161> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Turkmen_100 locale.

<162> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Uighur_PRC_100 locale.

<163> Section 2.324.4: SQL Server 2000 Reporting Services does not support the Ukrainian_100 locale.

<164> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Upper_Sorbian_100 locale.

<165> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Urdu_100 locale.

<166> Section 2.324.4: SQL Server 2000 Reporting Services does not support the Uzbek_Latin_90 locale.

<167> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Uzbek_Latin_100 locale.

<168> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Vietnamese_100 locale.

<169> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Welsh_100 locale.

<170> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Yakut_100 locale.

<171> Section 2.324.4: SQL Server 2000 Reporting Services and SQL Server 2005 Reporting Services do not support the Yi_100 locale.

<172> Section 2.324.13: 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 2.327.5: 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.Vaue.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 2.335: 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 2.335: 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 2.335.1: 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** 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 2.335.2: 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 2.335.3: 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 2.335.3: 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 2.337.1: 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 2.337.2: 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 2.339: 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 2.339: 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 2.339: 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 2.339: 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 2.339.2: 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 2.339.3: 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 2.340.3: 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 2.340.4: In the Microsoft implementation, the following table specifies how expression return types are converted to RDL data types.

RDL Type	CLR Types
String	**String**, **Char**, **GUID**
Boolean	**Boolean**
Integer	**Int16**, **Int32**, **Int64**, **UInt16**, **UInt32**, **UInt64**, **Byte**, **SByte**, **TimeSpan** The implementation of Reporting Services can convert **Integer** expression results to 32-bit or 64-bit precision, depending on the usage context.
DateTime	**DateTime**, **DateTimeOffset**
Float	**Single**, **Double**, **Decimal**
Binary	**Byte[]**
Variant	One of the following RDL types: **String**, **Boolean**, **Integer**, **DateTime**, **Float**, or **Binary**.
VariantArray	Array of **Variant**. Unless otherwise specified, in the Microsoft implementation, this is a true **variant** array (that is, object[]), not a strongly typed array.
StringArray	Array of **String**
Numeric	One of the following RDL types: **Integer** or **Float**.
Scalar	One of the following RDL types: **Integer**, **Float**, or **DateTime**.
Serializable	An RDL **Variant**, which is a CLR type that is marked with **Serializable** [ECMA-335], or a CLR type that implements **ISerializable**.

<190> Section 2.340.6.5: 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 2.340.6.6: 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 2.340.6.6: 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 2.340.6.12: The restriction of the use of the **DataSet** and **DataSource** collections in expressions within **Field** elements is not implemented in Reporting Services.

<194> Section 2.340.7.26.7: The Microsoft implementation allows strongly typed arrays in addition to object[].

<195> Section 2.340.7.26.7: 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 an 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.

Section	Description	Revision class
2 Structures	Updated links to RDL schemas.	Minor

9 Index

A

abstract base types 64
Action element 1110
ActionInfo element 1106
Aggregate function 1259
aggregate functions
  Aggregate function 1259
  Avg function 1253
  Count function 1254
  CountDistinct function 1254
  CountRows function 1255
  CreateDrillthroughContext function 1262
  filters 1260
  First function 1256
  Last function 1257
  list of 1250
  Max function 1253
  Min function 1254
  Previous function 1257
  recursive (section 2.340.7.2 1251, section 2.340.7.3 1252, section 2.340.7.4 1252)
  restrictions (section 2.340.7.23 1259, section 2.340.7.24 1260)
  RowNumber function 1258
  RunningValue function 1258
  scope (section 2.340.7.1 1251, section 2.340.7.26.1 1261)
  StDev function 1255
  StDevP function 1255
  Sum function 1252
  Union function 1256
  Var function 1256
  VarP function 1256
Applicability 71
Avg function 1253
Axis element 406

B

BandLayoutOptions element 260
Binary data type 1242
Body element 111
Boolean data type
  about 73
  expressions 1242
Border element 1153
BorderStyle element 1157
BorderWidth element 1162

C

calculations 62
CapImage element 828
CategoryAxis element 405
CategoryGrouping element 417
CategoryGroupings element 416
CellContents element 231
Change tracking 1647
Chart element 376
chart schema diagram 64
ChartAlignType element 471
ChartAnnotation element 463
ChartAnnotations element 463
ChartArea element 465
ChartAreas element 464
ChartAxis element 474
ChartCategoryAxes element 473
ChartData element 422
ChartDataLabel element 546
ChartDataPoint element 552
ChartDataPoints element 551
ChartDataPointValues element 558
ChartEmptyPoints element 567
ChartHierarchy element 586
ChartItemInLegend element 571
ChartLegend element 595
ChartLegends element 594
ChartMarker element 573
ChartMember element 589
ChartMembers element 588
ChartNoMoveDirections element 582
ChartSeries element 424
ChartSmartLabel element 575
Class element 1169
Classes element 1168
CodeModules element 1170
Column and Line Chart example 1279
ColumnGrouping element 302
ColumnGroupings element 301
ConnectionProperties element 1199
Corner element 312
Count function 1254
CountDistinct function 1254
CountRows function 1255
Coverflow element 263
CreateDrillthroughContext function 1262
custom assemblies
  in expressions 1242
Custom element 1114
custom report item schema diagram 64
CustomData element 1072
CustomLabel element 705
CustomLabels element 704
CustomProperties element 1115
CustomProperty element 1116
CustomReportItem element 1094
CustomReportItem example 1300

D

Data example 1326
data region scope 1251
data types
  Boolean 73
  DateTime 74
  expressions 1242
  Float 74
  in expressions 1241
  Integer 73
  NormalizedString 74
  String 73

DataCell element 1085
DataColumnGroupings element 1089
DataGrouping element 1090
DataGroupings element 1090
DataLabel element 439
DataPoint element 435
DataPoints element 434
DataRowGroupings element 1093
DataSet element 1202
dataset scope 1251
DataSetReference element 1186
DataSets element 1201
DataSource element 1197
DataSources element 1196
DataValues element 442
DateTime data type
  about 74
  expressions 1242
DefaultRelationship element 1234
DefaultRelationships element 1233
DefaultValue element 1185
Doughnut Chart example 1297
Drillthrough element 1113
dynamic scoping 1261
DynamicCategories element 418
DynamicColumns element 304
DynamicRows element 321
DynamicSeries element 451

E

EmbeddedImage element 1172
EmbeddedImages element 1171
errors
  error codes 1582
  expression error handling 1242
Examples 1266
  Column and Line Chart 1279
  CustomReportItem 1300
  Data 1326
  Doughnut Chart 1297
  GaugePanel 1330
  Line 1336
  List 1337
  Map 1343
  Matrix 1352
  Rectangle with Image and Textbox 1341
  Report in RDL schema 2005/01 1276
  Report in RDL schema 2008/01 1266
  Report in RDL schema 2010/01 1270
  Subreport 1327
  Table 1358
  Tablix 1 1305
  Tablix 2 1307
  Tablix 3 1313
  TextBox 1327
explicit scope 1251
Expressions (section 2.340 1241, section 2.340.1 1241)
  data types (section 2.340.1 1241, section 2.340.4 1242)
  error handling 1242
  global collections 1243
  referencing custom assemblies 1242
  referencing custom code methods 1242
  referencing function libraries 1242
  syntax 1241

F

Field element 1218
Fields - vendor-extensible 71
Fields element 1217
filters and aggregate functions 1260
Filters element 1221
FilterValues element 1225
First function 1256
Float data type
  about 74
  expressions 1242
functions
  aggregate (section 2.340.7 1250, section 2.340.7.23 1259, section 2.340.7.24 1260)
  Aggregate function 1259
  Avg function 1253
  Count function 1254
  CountDistinct function 1254
  CountRows function 1255
  CreateDrillthroughContext function 1262
  First function 1256
  InScope function 1261
  Last function 1257
  Level function 1252
  Lookup function 1262
  LookupSet function 1262
  Max function 1253
  Min function 1254
  MultiLookup function 1263
  Previous function 1257
  recursive (section 2.340.7.2 1251, section 2.340.7.3 1252)
  RowNumber function 1258
  RunningValue function 1258
  scope (section 2.340.7.1 1251, section 2.340.7.26.1 1261)
  semantic queries 1261
  StDev function 1255
  StDevP function 1255
  Sum function 1252
  Union function 1256
  Var function 1256
  VarP function 1256

G

gauge panel schema diagram 64
GaugeImage element 664
GaugeImages element 663
GaugeInputValue element 717
GaugeLabels element 672
GaugeMember element 680
GaugePanel element 636
GaugePanel example 1330
GaugeTickMarks 722
Glossary 55
Grouping element 368
grouping scope 1251

I

Image element 132
implicit scope 1251
IndicatorImage element 857
IndicatorState element 849
IndicatorStates element 848
Informative references 62
InScope function 1261
Integer data type
  about 73
  expressions 1242
Introduction 55

J

JoinCondition element 1236
JoinConditions element 1235

L

LabelData element 270
language code 86
Last function 1257
Level function 1252
Line element 143
Line example 1336
LinearGauge element 683
LinearGauges element 682
LinearScale element 692
LinearScales element 691
List element 272
List example 1337
Localization 71
LocIDStringWithDataTypeAttribute type 87
Lookup function 1262
LookupSet function 1262

M

Map example 1343
MapBindingFieldPair element 915
MapBindingFieldPairs element 914
MapBucket element 930
MapBuckets element 929
MapColorPaletteRule element 922
MapColorRangeRule element 931
MapColorScale element 876
MapColorScaleTitle element 887
MapCustomColorRule element 938
MapCustomColors elements 945
MapCustomView element 1061
MapDataBoundView element 1062
MapDataRegion element 893
MapElementView element 1063
MapFieldDefinition element 918
MapFieldDefinitions element 917
MapFields element 955
MapLegend element 1029
MapLegends element 1028
MapLegendTitle element 1039
MapLimits element 1065
MapLine element 952
MapLineRules element 920
MapLines element 951
MapLocation element 888
MapMarker element 991
MapMarkerImage element 993
MapMarkerRule element 985
MapMarkers element 990
MapMarkerTemplate element 975
MapMeridians element 1067
MapParallels element 1070
MapPointRules element 982
MapPolygonTemplate element 1015
MapSizeRule element 945
MapTile element 1027
MapTileLayer element 1022
MapTiles element 1026
MapTitle element 1043
MapTitles element 1042
MapViewport element 1050
Marker element 443
Matrix example 1352
MatrixCell element 317
MatrixCells element 317
MatrixColumn element 314
MatrixColumns element 313
MatrixRow element 315
MatrixRows element 314
Max function 1253
Min function 1254
MinorGridLines element 415
MultiLookup function 1263

N

Namespace 63
NavigationItem element 267
negative RdlSize 76
NormalizedString data type 74
Normative references 61
NumericIndicator element 775
NumericIndicators element 774

O

Overview (synopsis) 62

P

Page element 114
PageHeaderFooter element 123
PageSection element 120
Paragraph element 187
Paragraphs element 186
Parameter element 1165
Parameters element 1164
ParameterValue element 1191
ParameterValues element 1190
PlayAxis element 266
PlotArea element 448
PointerCap element 825
Previous function 1257
Product behavior 1633

Q

Query element 1227
QueryParameter element 1231
QueryParameters element 1230

R

RadialPointer element 815
RadialPointers element 814
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 150
Rectangle with Image and Textbox example 1341
recursive functions (section 2.340.7.2 1251, section 2.340.7.3 1252, section 2.340.7.4 1252)
References 61
  informative 62
  normative 61
Relationship element 1239
Relationship to protocols and other structures 70
Relationships element 1238
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 1276
Report in RDL schema 2008/01 example 1266
Report in RDL schema 2010/01 example 1270
report items 64
report layout (section 1.3.1 62, section 1.3.4 64)
report rendering 63
ReportItems element 126
ReportLanguage type 86
ReportMIMEType type 84
ReportParameters element 1174
ReportPath type 85
ReportSection element 107
ReportSections element 107
RowGrouping element 319
RowGroupings element 318
RowNumber function 1258
RunningValue function 1258

S

Scalar types 1242
ScalePin 751
ScaleRange element 763
ScaleRanges element 762
schema diagrams 64
scope
  about 1251
  dynamic scoping 1261
semantic queries 1261
SeriesGrouping element 450
SeriesGroupings element 449
Slider element 269
SortBy element 375
SortExpression element 365
SortExpressions element 364
Sorting element 374
StateImage element 855
StateIndicator element 833
StateIndicators element 832
StaticCategories element 420
StaticColumn element 311
StaticColumns element 310
StaticMember element 421
StaticRow element 325
StaticRows element 324
StaticSeries element 453
StDev function 1255
StDevP function 1255
String data type
  about 73
  expressions 1242
StringWithDataTypeAttribute type 86
Subreport element 161
Subreport example 1327
Subtotal element 307
Sum function 1252

T

Table element 326
Table example 1358
Tablix 1 example 1305
Tablix 2 example 1307
Tablix 3 example 1313
Tablix element 200
tablix schema diagram 64
TablixBody element 221
TablixCell element 228
TablixCells element 227
TablixColumn element 223
TablixColumnHierarchy element 239
TablixColumns element 222
TablixCorner element 253
TablixCornerCell element 256
TablixCornerRow element 255
TablixCornerRows element 254
TablixHeader element 251
TablixHierarchy element 257
TablixMember element 242
TablixMembers element 241
TablixRow element 225
TablixRowHierarchy element 259
TablixRows element 224
Tabstrip element 265
Textbox element 173
TextBox example 1327
TextRun element 193
TextRuns element 192
ThreeDProperties element 453
TickMarkStyle element 711
Title element 460
ToggleImage element 197
TopImage element 770
Tracking changes 1647
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 1256
UserSort element 198

V

ValidValues element 1189
ValueAxis element 462
Values element 1188
Var function 1256
Variable element 1194
Variables element 1193
VariantArray 1242
VarP function 1256
Vendor-extensible fields 71
Versioning 71
Visibility element 1166

X

XML namespace 63
XML schema diagrams 64