

# [MS-RDLRS]: Report Definition Language Report State File Format Specification

---

## Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **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 may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft [Open Specification Promise](#) or the [Community Promise](#). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events 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 specifically described above, whether by implication, estoppel, or otherwise.

**Tools.** The Open Specifications do 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 are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

## Revision Summary

Date	Revision History	Revision Class	Comments
11/03/2011	1.0	New	Released new document.
01/19/2012	1.0	No change	No changes to the meaning, language, or formatting of the technical content.
02/23/2012	2.0	Major	Significantly changed the technical content.

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Glossary	5
1.2	References	5
1.2.1	Normative References	5
1.2.2	Informative References	6
1.3	Overview	6
1.3.1	Document Structure	6
1.3.1.1	XML Namespace	6
1.4	Relationship to Protocols and Other Structures	6
1.5	Applicability Statement	7
1.6	Versioning and Localization	7
1.7	Vendor-Extensible Fields	7
<b>2</b>	<b>Structures</b>	<b>8</b>
2.1	Common Report State Types	8
2.1.1	Boolean	8
2.1.2	Byte	8
2.1.3	DateTime	8
2.1.4	Decimal	8
2.1.5	Double	8
2.1.6	Guid	8
2.1.7	Single	8
2.1.8	SByte	9
2.1.9	Int16	9
2.1.10	Int32	9
2.1.11	Int64	9
2.1.12	String	9
2.1.13	Time	9
2.1.14	EntityReference	9
2.1.15	PropertyReference	10
2.2	ReportState	10
2.3	ReportSections	10
2.4	ReportSection	11
2.5	SectionFilters	11
2.6	Filters	11
2.7	Filter	12
2.8	PrimitiveType	13
2.9	FilterCondition	14
2.10	SimpleFilterCondition	14
2.11	CompoundFilterCondition	16
2.12	ReportItems	17
2.13	DataRegion	17
2.14	TypeUsage	17
2.15	QueryExpressionBinding	18
2.16	QueryExpressionList	18
2.17	QueryExpression	18
2.18	QueryCalculateExpression	19
2.19	QueryFieldExpression	19
2.20	QueryFunctionExpression	20
2.21	QueryLiteralExpression	21

2.22	QueryMeasureExpression .....	22
2.23	QueryNullExpression .....	22
2.24	QueryProjectExpression .....	23
2.25	QueryScanExpression .....	23
2.26	QueryVariableReferenceExpression.....	24
2.27	QdmEntityPlaceholderExpression .....	24
2.28	ReportSlicerState .....	24
<b>3</b>	<b>Structure Examples .....</b>	<b>26</b>
3.1	Report State .....	26
<b>4</b>	<b>Security.....</b>	<b>31</b>
4.1	Security Considerations for Implementers.....	31
<b>5</b>	<b>Appendix A: Full XML Schema .....</b>	<b>32</b>
5.1	Report State Schema .....	32
5.2	Serialization Schema.....	37
<b>6</b>	<b>Appendix C: Product Behavior .....</b>	<b>39</b>
<b>7</b>	<b>Change Tracking.....</b>	<b>40</b>
<b>8</b>	<b>Index .....</b>	<b>42</b>

# 1 Introduction

This document specifies the file format for Report State. A Report State document, that is, a document in the Report State file format, represents the metadata for defining **filter** in a **report** [\[MS-RDL\]](#) and storing information about interactions with those filters.

Sections 1.7 and 2 of this specification are normative and contain RFC 2119 language. All other sections and examples in this specification are informative.

## 1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

**globally unique identifier (GUID)**  
**report definition**

The following terms are defined in [\[MS-RDL\]](#):

**data region**  
**filter**  
**report**  
**report item**  
**report section**

The following terms are defined in [\[MC-CSDL\]](#):

**entity**  
**property**

The following terms are specific to this document:

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

## 1.2 References

References to Microsoft Open Specification documents do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

### 1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[MC-CSDL] Microsoft Corporation, "[Conceptual Schema Definition File Format](#)".

[MS-RDL] Microsoft Corporation, "[Report Definition Language File Format Structure Specification](#)".

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

[XML1.0/5] Bray, T., Maler, E., Paoli, J., "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.S., Ed., Beech, D., Ed., Maloney, M., Ed., and Mendelsohn, N., Ed., "XML Schema Part 1: Structures Second Edition", W3C Recommendation, October 2004, <http://www.w3.org/TR/xmlschema-1/>

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

## 1.2.2 Informative References

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)".

## 1.3 Overview

This document specifies the file format for Report State. A Report State document, that is, a document in the Report State file format, represents the metadata for defining filters in a report [\[MS-RDL\]](#) and storing information about interactions with those filters. A Report State document is an XML file, as specified in [\[XML1.0/5\]](#).

### 1.3.1 Document Structure

The root element of a Report State document is a [ReportState](#) element. Report State documents describe metadata about filters that apply to **report sections** and/or **data regions**.

#### 1.3.1.1 XML Namespace

The namespace URI for Report State is as follows:

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

The date component (yyyy/mm) indicates the date of the release of that version of Report State. The date component is used throughout this specification to identify the following specific version of Report State:

- 2011/01: [Appendix A: Full XML Schema](#)

## 1.4 Relationship to Protocols and Other Structures

Report State documents represent filter metadata for an RDL [\[MS-RDL\]](#) file. Several cross-references exist between [\[MS-RDL\]](#) and the structures that are defined in this specification:

- [ReportSection](#) structures within a Report State document correspond to report section structures in an RDL file.
- [DataRegion](#) structures within a Report State document refer to data regions defined in an RDL file.
- Some items in an RDL file can contain a [ReportSlicerState](#) structure.

Report State files and structures are interpreted in the context of a Conceptual Schema document [\[MC-CSDL\]](#). Some structures defined in this specification include references to **entity** containers, entity sets, or **properties** that are defined in a Conceptual Schema document as specified in [\[MC-CSDL\]](#).

## 1.5 Applicability Statement

The Report State file format is applicable for use in representing the filter metadata for a **report definition**. Report State documents do not exist in isolation. Report State documents accompany RDL documents, specified in [MS-RDL], to describe filter metadata.

## 1.6 Versioning and Localization

This document covers versioning issues in the following areas:

- 2011/01: This document describes the 2011/01 version of Report State structures. This is the only version of these structures.

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

## 1.7 Vendor-Extensible Fields

The Report State file format contains no vendor-extensible fields or structures.

## 2 Structures

### 2.1 Common Report State Types

#### 2.1.1 Boolean

If the Report State 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 **xs:Boolean**.

#### 2.1.2 Byte

If the Report State format specifies an element or attribute as type **Byte**, it MUST be a valid **unsignedByte** value, as specified in [\[XMLSCHEMA2/2\]](#) section 3.3.24 for **xs:unsignedByte**.

#### 2.1.3 DateTime

If the Report State format specifies an element or attribute as type **DateTime**, it MUST be a valid date and time-of-day value as specified in [\[XMLSCHEMA2/2\]](#) section 3.2.7 for **xs:dateTime**.

#### 2.1.4 Decimal

If the Report State format specifies an element or attribute as type **Decimal**, it MUST be a valid **decimal** value, as specified in [\[XMLSCHEMA2/2\]](#) section 3.2.3 for **xs:decimal**.

#### 2.1.5 Double

If the Report State format specifies an element or attribute as type **Double**, it MUST be a valid **double** value, as specified in [\[XMLSCHEMA2/2\]](#) section 3.2.5 for **xs:double**.

#### 2.1.6 Guid

The **guid** simple type is from namespace <http://schemas.microsoft.com/2003/10/Serialization/>.

The **guid** simple type represents a **globally unique identifier (GUID)**.

The following is the XSD for the **guid** simple type.

```
<xs:simpleType name="guid">
  <xs:restriction base="xs:string">
    <xs:pattern value="\{8\}-[4]-[4]-[4]-[4]-[12]"/>
  </xs:restriction>
</xs:simpleType>
```

#### 2.1.7 Single

If the Report State format specifies an element or attribute as type **Single**, it MUST be a valid **single** value, as specified in [\[XMLSCHEMA2/2\]](#) section 3.2.4 for **xs:float**.

### 2.1.8 SByte

If the Report State format specifies an element or attribute as type **SByte**, it MUST be a valid **byte** value as specified in [\[XMLSCHEMA2/2\]](#) section 3.3.19 for **xs:byte**.

### 2.1.9 Int16

If the Report State format specifies an element or attribute as type **Int16**, it MUST be a valid **short** value as specified in [\[XMLSCHEMA2/2\]](#) section 3.3.18 for **xs:short**.

### 2.1.10 Int32

If the Report State format specifies an element or attribute as type **Int32**, it MUST be a valid **int** value as specified in [\[XMLSCHEMA2/2\]](#) section 3.3.17 for **xs:int**.

### 2.1.11 Int64

If the Report State format specifies an element or attribute as type **Int64**, it MUST be a valid **long** value as specified in [\[XMLSCHEMA2/2\]](#) section 3.3.16 for **xs:long**.

### 2.1.12 String

If the Report State format specifies an element or attribute as type **String**, it MUST be a valid **string** value as specified in [\[XMLSCHEMA2/2\]](#) section 3.2.1 for **xs:string**.

### 2.1.13 Time

The **Time** simple type is from namespace <http://schemas.microsoft.com/2003/10/Serialization/>.

The **Time** simple type specifies time duration.

The following is the XSD for the **Time** simple type.

```
<xs:element name="duration" nillable="true" type="tns:duration"/>
<xs:simpleType name="duration">
  <xs:restriction base="xs:duration">
    <xs:pattern
      value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?"/>
    <xs:minInclusive value="-P10675199DT2H48M5.4775808S"/>
    <xs:maxInclusive value="P10675199DT2H48M5.4775807S"/>
  </xs:restriction>
</xs:simpleType>
```

### 2.1.14 EntityReference

If the Report State format defines an element or attribute as type **EntityReference**, it MUST be a valid string as specified in [\[XMLSCHEMA2/2\]](#). Additionally, the string MUST be of the format *EntityContainerName.EntitySetName*.

- *EntityContainerName* refers to the value of the **Name** attribute on an **EntityContainer** element in a CSDL document [\[MC-CSDL\]](#).
- *EntitySetName* refers to the value of the **Name** attribute of an **EntitySet** element in a CSDL document that is contained within the specified **EntityContainer** [\[MC-CSDL\]](#).

### 2.1.15 PropertyReference

If the Report State format defines an element or attribute as type **PropertyReference**, it MUST be a valid string as specified in [XMLSCHEMA2/2](#). Additionally, the string refers to the name of a property defined on an **EntitySet** element in a CSDL document [MC-CSDL](#).

## 2.2 ReportState

The **ReportState** element describes the state created through user interactions with a report [MS-RDL](#).

**FilterAreaVisibility:** Specifies an element that describes the state of the user interface that is used to display filters. The **FilterAreaVisibility** element is optional and MUST NOT be specified more than once.

The values and meanings for **FilterAreaVisibility** are as follows.

Value	Meaning
Collapsed	The filter area is visible in a collapsed state.
Expanded	The filter area is visible in an expanded state.
Hidden	The filter area is not displayed.

**Theme:** Specifies an element that describes the name of the theme that is used by the report. The **Theme** element is optional and MUST NOT be specified more than once.

**ReportSections:** Specifies a [ReportSections](#) element that specifies the state for items within that section. The **ReportSections** element is optional and MUST NOT be specified more than once.

The following is the XSD for the **ReportState** element.

```
<xs:element name="ReportState">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="FilterAreaVisibility" minOccurs="0">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="Collapsed"/>
            <xs:enumeration value="Expanded"/>
            <xs:enumeration value="Hidden"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:element name="Theme" minOccurs="0"/>
      <xs:element name="ReportSections" minOccurs="0" type="ReportSectionsType" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

## 2.3 ReportSections

The **ReportSections** element specifies an ordered list of [ReportSection](#) elements.

**ReportSection:** Specifies a **ReportSection** that describes the state of each section within a report. The **ReportSection** element is optional and MAY be specified more than once.

The following is the XSD for the **ReportSections** element.

```
<xs:complexType name="ReportSectionsType">
  <xs:sequence>
    <xs:element name="ReportSection" type="ReportSectionType" minOccurs="0"
maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

## 2.4 ReportSection

The **ReportSection** element describes the state of each report section in a report definition.

**SectionFilters:** Specifies a [SectionFilters](#) element that describes the state of filters that are defined for this section of the report. The **SectionFilters** element is optional and MUST NOT be specified more than once.

**ReportItems:** Specifies a [ReportItems](#) element that describes the state of the **report items** within this section. The **ReportItems** element is optional and MUST NOT be specified more than once.

The following is the XSD for the **ReportSection** element.

```
<xs:complexType name="ReportSectionType">
  <xs:sequence>
    <xs:element name="SectionFilters" type="SectionFiltersType" minOccurs="0" />
    <xs:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
  </xs:sequence>
</xs:complexType>
```

## 2.5 SectionFilters

The **SectionFilters** element is an ordered list of [Filters](#) elements.

**Filters:** Specifies a **Filters** element that describes the states of the filter within a [ReportSection](#). The **Filters** element MUST be specified and MUST NOT be specified more than once.

The following is the XSD for the **SectionFilters** element.

```
<xs:complexType name="SectionFiltersType">
  <xs:sequence>
    <xs:element name="Filters" type="FiltersType" />
  </xs:sequence>
</xs:complexType>
```

## 2.6 Filters

The **Filters** element is an ordered list of [Filter](#) elements.

**Filter:** Specifies a **Filter** that describes the state of a filter. The **Filter** element MUST be specified at least once.

The following is the XSD for the **Filters** element.

```
<xs:complexType name="FiltersType">
  <xs:sequence>
    <xs:element name="Filter" type="FilterType" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

## 2.7 Filter

The **Filter** element describes the state of a filter within a [ReportSection](#) or a [DataRegion](#).

**Mode:** Specifies an element that describes the state of the filter's mode.

The values and meanings for the **Mode** element are as follows.

Value	Meaning
Advanced	The filter is in advanced mode.
Basic	The filter is in basic mode.
Cleared	The filter is cleared.

**IsMeasure:** Specifies whether this filter is applied to aggregated data. The **IsMeasure** element is optional and MUST NOT be specified more than once.

**Operand:** Specifies an element that describes the state of an operand that was used in a filter. The **Operand** element MUST be specified and MUST NOT be specified more than once.

**Operand.QueryExpression:** Specifies a [QueryExpression](#) that was used as the filter operand. The **Operand.QueryExpression** element MUST be specified and MUST NOT be specified more than once.

**Condition:** Specifies an element that describes the state of a condition in a filter. The **Condition** element is optional and MUST NOT be specified more than once.

**Condition.CompoundFilterCondition:** Specifies a [CompoundFilterCondition](#) that describes a set of conditions that are logically combined through a filter operator. The **Condition.CompoundFilterCondition** element MUST be specified and MUST NOT be specified more than once.

**Type:** Specifies the [PrimitiveType](#) of the data being filtered. The **Type** attribute MUST be specified and MUST NOT be specified more than once.

Following is the XSD for the **Filter** element.

```
<xs:complexType name="FilterType">
  <xs:sequence>
    <xs:element name="Mode">
      <xs:simpleType>
```

```

    <xs:restriction base="xs:string">
      <xs:enumeration value="Advanced" />
      <xs:enumeration value="Basic" />
      <xs:enumeration value="Cleared" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="IsMeasure" minOccurs="0" />
<xs:element name="Operand">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="QueryExpression" type="QueryExpression"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Condition" minOccurs="0">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CompoundFilterCondition" type="CompoundFilterCondition"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="Type" type="PrimitiveType" />
</xs:complexType>

```

## 2.8 PrimitiveType

The **PrimitiveType** element specifies the name of a data type.

The values and meanings for the **PrimitiveType** element are as follows.

Value	Meaning
Boolean	Specifies use of the <a href="#">Boolean</a> data type.
Byte	Specifies use of the <a href="#">Byte</a> data type.
DateTime	Specifies use of the <a href="#">DateTime</a> data type.
Decimal	Specifies use of the <a href="#">Decimal</a> data type.
Double	Specifies use of the <a href="#">Double</a> data type.
Guid	Specifies use of the <a href="#">Guid</a> data type.
Single	Specifies use of the <a href="#">Single</a> data type.
SByte	Specifies use of the <a href="#">SByte</a> data type.
Int16	Specifies use of the <a href="#">Int16</a> data type.
Int32	Specifies use of the <a href="#">Int32</a> data type.
Int64	Specifies use of the <a href="#">Int64</a> data type.
String	Specifies use of the <a href="#">String</a> data type.

Value	Meaning
Time	Specifies use of the <a href="#">Time</a> data type.

The following is the XSD for the **PrimitiveType** element.

```
<xs:simpleType name="PrimitiveType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Boolean" />
    <xs:enumeration value="Byte" />
    <xs:enumeration value="DateTime" />
    <xs:enumeration value="Decimal" />
    <xs:enumeration value="Double" />
    <xs:enumeration value="Guid" />
    <xs:enumeration value="Single" />
    <xs:enumeration value="SByte" />
    <xs:enumeration value="Int16" />
    <xs:enumeration value="Int32" />
    <xs:enumeration value="Int64" />
    <xs:enumeration value="String" />
    <xs:enumeration value="Time" />
  </xs:restriction>
</xs:simpleType>
```

## 2.9 FilterCondition

The **FilterCondition** element is an abstract element that represents a condition that is used to filter a set of values. In Report State documents, elements of type **FilterCondition** MUST specify one of the following types by using a type attribute as defined in [\[XMLSCHEMA1/2\]](#) section 2.6.1.

Type
<a href="#">SimpleFilterCondition</a>
<a href="#">CompoundFilterCondition</a>

The following is the XSD for the **FilterCondition** element.

```
<xs:complexType name="FilterCondition" abstract="true" />
```

## 2.10 SimpleFilterCondition

The **SimpleFilterCondition** element specifies a type of [FilterCondition](#) that relates a **LeftExpression** and **RightExpression**.

**LeftExpression:** Specifies the [QueryExpression](#) that is on the left side of the **Operator**. The **LeftExpression** element MUST be specified and MUST NOT be specified more than once.

**Not:** Specifies whether this filter condition should be logically negated. The **Not** element is optional and MUST NOT be specified more than once.

**Operator:** Specifies the filter operation to perform on the **LeftExpression** and the **RightExpression**. The **Operator** element MUST be specified and MUST NOT be specified more than once.

The values and meanings for the **Operator** element are as follows.

Value	Meaning
Equal	The value of <b>LeftExpression</b> and the value of <b>RightExpression</b> are equal.
GreaterThan	The value of <b>LeftExpression</b> is greater than the value of <b>RightExpression</b> .
GreaterThanOrEqual	The value of <b>LeftExpression</b> is greater than or equal to the value of <b>RightExpression</b> .
LessThan	The value of <b>LeftExpression</b> is less than the value of <b>RightExpression</b> .
LessThanOrEqual	The value of <b>LeftExpression</b> is less than or equal to the value of <b>RightExpression</b> .
Contains	The value of <b>LeftExpression</b> contains the value of <b>RightExpression</b> .
StartsWith	The value of <b>LeftExpression</b> starts with the value of <b>RightExpression</b> .
DateTimeEqualToSecond	The value of <b>LeftExpression</b> equals the value of <b>RightExpression</b> when any sub-second precision of the value of <b>LeftExpression</b> is ignored. The <b>RightExpression</b> MUST be a QueryLiteralExpression.

**RightExpression:** Specifies the **QueryExpression** element that is on the right side of the **Operator**. The **RightExpression** element is optional and MUST NOT be specified more than once.

The following is the XSD for the **SimpleFilterCondition** element.

```
<xs:complexType name="SimpleFilterCondition">
  <xs:complexContent>
    <xs:extension base="FilterCondition">
      <xs:sequence>
        <xs:element name="LeftExpression" type="QueryExpression" />
        <xs:element name="Not" type="xs:boolean" minOccurs="0" />
        <xs:element name="Operator">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="Equal" />
              <xs:enumeration value="GreaterThan" />
              <xs:enumeration value="GreaterThanOrEqual" />
              <xs:enumeration value="LessThanOrEqual" />
              <xs:enumeration value="LessThan" />
              <xs:enumeration value="Contains" />
              <xs:enumeration value="StartsWith" />
              <xs:enumeration value="DateTimeEqualToSecond" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="RightExpression" type="QueryExpression" minOccurs="0" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

## 2.11 CompoundFilterCondition

The **CompoundFilterCondition** element specifies a type of [FilterCondition](#) that combines other **FilterCondition** elements.

**Conditions:** Specifies the ordered set of **FilterCondition** elements to be logically combined together. This **Conditions** element is optional and MUST NOT be specified more than once.

**Conditions.FilterCondition:** Specifies a **FilterCondition**. The **Conditions.FilterCondition** element is optional and MAY be specified more than once.

**Operator:** Specifies the desired logical combination of the **Conditions**. The **Operator** element is optional and MUST NOT be specified more than once.

The values and meanings for the **Operator** element are as follows.

Value	Meaning
All	All of the <b>Conditions</b> evaluate to true.
Any	Any of the <b>Conditions</b> evaluates to true.
NotAll	Any of the <b>Conditions</b> evaluates to false.
NotAny	All of the <b>Conditions</b> evaluate to false.

The following is the XSD for the **CompoundFilterCondition** element.

```
<xs:complexType name="CompoundFilterCondition">
  <xs:complexContent>
    <xs:extension base="FilterCondition">
      <xs:sequence>
        <xs:element name="Conditions" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="FilterCondition" type="FilterCondition" minOccurs="0"
maxOccurs="unbounded" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Operator" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="All" />
              <xs:enumeration value="Any" />
              <xs:enumeration value="NotAll" />
              <xs:enumeration value="NotAny" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

## 2.12 ReportItems

The **ReportItems** element specifies an ordered set of [DataRegion](#) elements.

**DataRegion:** Specifies a **DataRegion**. This **DataRegion** element MUST be specified and MAY be specified more than once.

The following is the XSD for the **ReportItems** element.

```
<xs:complexType name="ReportItemsType">
  <xs:sequence>
    <xs:element name="DataRegion" type="DataRegionType" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```

## 2.13 DataRegion

The **DataRegion** element specifies the state of a named data region in a report definition.

**Filters:** Specifies the container element for the [Filters](#). The **Filters** element MUST be specified and MUST NOT be specified more than once.

**Filters.Filters:** Specifies the **Filters** to use. The **Filters.Filters** element MUST be specified and MUST NOT be specified more than once.

**Name:** Specifies the name of the data region where the **Filters** are applied. The **Name** attribute MUST be specified and MUST NOT be specified more than once.

The following is the XSD for the **DataRegion** element.

```
<xs:complexType name="DataRegionType">
  <xs:sequence>
    <xs:element name="Filters">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Filters" type="FiltersType" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="Name" type="xs:string" />
</xs:complexType>
```

## 2.14 TypeUsage

The **TypeUsage** element specifies a particular use of a primitive data type.

**PrimitiveType:** Specifies the [PrimitiveType](#) element to use. The **PrimitiveType** element MUST be specified and MUST NOT be specified more than once.

The following is the XSD for the **TypeUsage** element.

```

<xs:complexType name="TypeUsage">
  <xs:sequence>
    <xs:element name="PrimitiveType" type="PrimitiveType" />
  </xs:sequence>
</xs:complexType>

```

## 2.15 QueryExpressionBinding

The **QueryExpressionBinding** element specifies a loop over the set of values described by the **Expression** element.

**Expression:** Specifies the set of values being enumerated. The **Expression** element MUST be specified and MUST NOT be specified more than once.

**VariableName:** Specifies the name used to refer to the current value while iterating over the set of values. The **VariableName** element MUST be specified and MUST NOT be specified more than once.

The following is the XSD for the **QueryExpressionBinding** element.

```

<xs:complexType name="QueryExpressionBinding">
  <xs:sequence>
    <xs:element name="Expression" type="QueryExpression"/>
    <xs:element name="VariableName" type="xs:string" />
  </xs:sequence>
</xs:complexType>

```

## 2.16 QueryExpressionList

The QueryExpressionList element specifies an ordered list of [QueryExpression](#) elements.

**QueryExpression:** Specifies a **QueryExpression** element. This element is optional and MAY be specified more than once.

The following is the XSD for the **QueryExpressionList** element.

```

<xs:complexType name="QueryExpressionList">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="QueryExpression" type="QueryExpression"/>
  </xs:sequence>
</xs:complexType>

```

## 2.17 QueryExpression

**QueryExpression** is an abstract element. In Report State documents, **QueryExpression** elements MUST specify one of the following types by using a type attribute as defined in [\[XMLSCHEMA1/2\]](#) section 2.6.1.

Type
<a href="#">QueryCalculateExpression</a>

Type
<a href="#">QueryFieldExpression</a>
<a href="#">QueryFunctionExpression</a>
<a href="#">QueryLiteralExpression</a>
<a href="#">QueryMeasureExpression</a>
<a href="#">QueryNullExpression</a>
<a href="#">QueryProjectExpression</a>
<a href="#">QueryScanExpression</a>
<a href="#">QueryVariableReferenceExpression</a>
<a href="#">QdmEntityPlaceholderExpression</a>

The following is the XSD for the **QueryExpression** element.

```
<xs:complexType name="QueryExpression" abstract="true" />
```

## 2.18 QueryCalculateExpression

The **QueryCalculateExpression** element specifies a [QueryExpression](#) element that should be evaluated in a context that is intersected with the specified filter expressions. The **QueryCalculateExpression** element is of type **QueryExpression**.

**Argument:** Specifies the **QueryExpression** element to be evaluated. The **Argument** element MUST be specified and MUST NOT be specified more than once.

**Filters:** Specifies the **QueryExpression** element that is used to filter data while evaluating to the argument. The [Filters](#) element is optional and MUST NOT be specified more than once.

The following is the XSD for the **QueryCalculateExpression** element.

```
<xs:complexType name="QueryCalculateExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="Argument" type="QueryExpression" />
        <xs:element name="Filters" type="QueryExpressionList" minOccurs="0" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

## 2.19 QueryFieldExpression

The **QueryFieldExpression** element specifies a reference to a named value projected from the **Instance** [QueryExpression](#). The **QueryFieldExpression** element is of type **QueryExpression**.

**Field:** Specifies the name of the value to be retrieved. This element MUST be of type [PropertyReference](#). The **Field** element MUST be specified and MUST NOT be specified more than once.

**Instance:** Specifies the **QueryExpression** from which the value should be retrieved. The **Instance** element MUST be specified and MUST NOT be specified more than once.

The following is the XSD for the **QueryFieldExpression** element.

```
<xs:complexType name="QueryFieldExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="Field" type="xs:string" />
        <xs:element name="Instance" type="QueryExpression" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

## 2.20 QueryFunctionExpression

The QueryFunctionExpression element specifies the function to invoke over a set of arguments as input. The **QueryFunctionExpression** element is of type [QueryExpression](#).

**FunctionName:** Specifies the name of the function to invoke. The **FunctionName** element MUST be specified and MUST NOT be specified more than once.

The values and meanings for the **FunctionName** element are as follows.

Value	Meaning
<b>Core.Sum</b>	Determines the aggregate summation of all values of the one and only argument. <b>QueryFunctionExpression.Arguments</b> MUST be of size one.
<b>Core.Count</b>	Determines the aggregate count of all non-null values of the one and only argument. <b>QueryFunctionExpression.Arguments</b> MUST be of size one.
<b>Core.DistinctCount</b>	Determines the aggregate count of distinct non-null values of the one and only argument. <b>QueryFunctionExpression.Arguments</b> MUST be of size one.
<b>Core.Average</b>	Determines the aggregate average of all values of the one and only argument. <b>QueryFunctionExpression.Arguments</b> MUST be of size one.
<b>Core.Min</b>	Determines the aggregate minimum of all values of the one and only argument. <b>QueryFunctionExpression.Arguments</b> MUST be of size one.
<b>Core.Max</b>	Determines the aggregate maximum of all values of the one and only argument. <b>QueryFunctionExpression.Arguments</b> MUST be of size one.

**Arguments:** Specifies a [QueryExpressionList](#) element as the input arguments to the function. The **Arguments** element MUST be specified and MUST NOT be specified more than once.

The following is the XSD for the **QueryFunctionExpression** element.

```
<xs:complexType name="QueryFunctionExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="FunctionName">
          <xs:simpleType name="FunctionName">
            <xs:restriction base="xs:string">
              <xs:enumeration value="Core.Sum" />
              <xs:enumeration value="Core.Count" />
              <xs:enumeration value="Core.DistinctCount" />
              <xs:enumeration value="Core.Average" />
              <xs:enumeration value="Core.Min" />
              <xs:enumeration value="Core.Max" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Arguments" type="QueryExpressionList" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

## 2.21 QueryLiteralExpression

The QueryLiteralExpression element specifies a typed constant value. The **QueryLiteralExpression** element is of type [QueryExpression](#).

**Value:** Specifies the value of the constant. The **Value** element MUST be specified and MUST NOT be specified more than once. In Report State documents, use of the **Value** element MUST specify one of the following types by using a type attribute as defined in [\[XMLSCHEMA1/2\]](#) section 2.6.1.

Value
<a href="#">Boolean</a>
<a href="#">Byte</a>
<a href="#">DateTime</a>
<a href="#">Decimal</a>
<a href="#">Double</a>
<a href="#">Guid</a>
<a href="#">Single</a>
<a href="#">SByte</a>
<a href="#">Int16</a>
<a href="#">Int32</a>
<a href="#">Int64</a>

Value
<a href="#">String</a>
<a href="#">Time</a>

The following is the XSD for the **QueryLiteralExpression** element.

```
<xs:complexType name="QueryLiteralExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="Value" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

## 2.22 QueryMeasureExpression

The QueryMeasureExpression element specifies a reference to a named property on an entity. The **QueryMeasureExpression** element is of type [QueryExpression](#).

**Measure:** Specifies the name of the **property**. The **Measure** element MUST be of type [PropertyReference](#). The **Measure** element MUST be specified and MUST NOT be specified more than once.

**Target:** Specifies the target **entity** containing the item described by the **Measure** element. The **Target** element MUST be of type [EntityReference](#). The **Target** element MUST be specified and MUST NOT be specified more than once.

The following is the XSD for the **QueryMeasureExpression** element.

```
<xs:complexType name="QueryMeasureExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="Measure" type="xs:string" />
        <xs:element name="Target" type="xs:string" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

## 2.23 QueryNullExpression

The QueryNullExpression element specifies a null value in a data type. The **QueryNullExpression** element is of type [QueryExpression](#).

**Type:** Specifies the data type. The **Type** element MUST be specified and MUST NOT be specified more than once.

The following is the XSD for the **QueryNullExpression** element.

```

<xs:complexType name="QueryNullExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="Type" type="TypeUsage" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

## 2.24 QueryProjectExpression

The **QueryProjectExpression** element specifies a type of [QueryExpression](#) element that produces a value for each item in the specified input.

**Input:** Specifies the input [QueryExpressionBinding](#). The **Input** element MUST be specified and MUST NOT be specified more than once.

**Projection:** Specifies the **QueryExpression** element that produces the value. The **Projection** element MUST be specified and MUST NOT be specified more than once.

The following is the XSD for the **QueryProjectExpression** element.

```

<xs:complexType name="QueryProjectExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="Input" type="QueryExpressionBinding" />
        <xs:element name="Projection" type="QueryExpression" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

## 2.25 QueryScanExpression

The **QueryScanExpression** element specifies a scan over all rows in the target entity. The **QueryScanExpression** element is of type [QueryExpression](#).

**Target:** Specifies the entity. The **Target** element MUST be of type [EntityReference](#). The **Target** element MUST be specified and MUST NOT be specified more than once.

**ExcludeBlankRow:** Specifies a [Boolean](#) value that indicates whether blank rows should be excluded in the row scan. If omitted, blank rows are included. The **ExcludeBlankRow** element is optional and MUST NOT be specified more than once.

The following is the XSD for the **QueryScanExpression** element.

```

<xs:complexType name="QueryScanExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>

```

```

    <xs:element name="Target" type="xs:string" />
    <xs:element name="ExcludeBlankRow" type="xs:boolean" minOccurs="0" />
  </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

```

## 2.26 QueryVariableReferenceExpression

The **QueryVariableReferenceExpression** element specifies a reference to a named value. The **QueryVariableReferenceExpression** element is of type [QueryExpression](#).

**VariableName:** Specifies the name of the value. The **VariableName** element MUST be specified.

The following is the XSD for the **QueryVariableReferenceExpression** element.

```

<xs:complexType name="QueryVariableReferenceExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="VariableName" type="xs:string" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

## 2.27 QdmEntityPlaceholderExpression

The **QdmEntityPlaceholderExpression** element specifies a reference to an entity. The **QdmEntityPlaceholderExpression** element is of type [QueryExpression](#).

**Target:** Specifies the entity. The **Target** element MUST be of type **EntityReference**. The **Target** element MUST be specified and MUST NOT be specified more than once.

The following is the XSD for the **QdmEntityPlaceholderExpression** element.

```

<xs:complexType name="QdmEntityPlaceholderExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="Target" type="xs:string" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

## 2.28 ReportSlicerState

The **ReportSlicerState** element specifies a [CompoundFilterCondition](#) to be applied to items within a report section [\[MS-RDL\]](#). The **ReportSlicerState** element is not used directly within a Report State document.

**CompoundFilterCondition:** Specifies the filter condition to apply to items within a **report section**. The **CompoundFilterCondition** element **MUST** be specified and **MUST NOT** be specified more than once.

The following is the XSD for the **ReportSlicerState** element.

```
<xs:element name="ReportSlicerState">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CompoundFilterCondition" type="CompoundFilterCondition"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

### 3 Structure Examples

The following sections provide examples of the structures that are defined in the Report State file format.

#### 3.1 Report State

The following is an example of a Report State document with two [ReportSection](#) elements. The first **ReportSection** contains two section filters and two [DataRegion](#) elements with filters. The second section contains a single section filter and a single **DataRegion** with filters.

```
<?xml version="1.0" encoding="utf-8"?>
<ReportState xmlns="http://schemas.microsoft.com/sqlserver/reporting/2011/01/reportstate">
  <FilterAreaVisibility>Expanded</FilterAreaVisibility>
  <Theme>Theme1</Theme>
  <ReportSections>
    <ReportSection>
      <SectionFilters>
        <Filters
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2011/01/reportstate">
          <Filter Type="String">
            <Mode>Basic</Mode>
            <Operand>
              <QueryExpression xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
i:type="QueryFieldExpression">
                <Field>Genre</Field>
                <Instance i:type="QdmEntityPlaceholderExpression">
                  <Target>Sandbox.Media</Target>
                </Instance>
              </QueryExpression>
            </Operand>
          <Condition>
            <CompoundFilterCondition xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
              <Conditions>
                <FilterCondition i:type="SimpleFilterCondition">
                  <LeftExpression i:type="QueryFieldExpression">
                    <Field>Genre</Field>
                    <Instance i:type="QdmEntityPlaceholderExpression">
                      <Target>Sandbox.Media</Target>
                    </Instance>
                  </LeftExpression>
                  <Operator>Equal</Operator>
                  <RightExpression i:type="QueryLiteralExpression">
                    <Value xmlns:d5p1="http://www.w3.org/2001/XMLSchema"
i:type="d5p1:string">Action/Adventure</Value>
                  </RightExpression>
                </FilterCondition>
              </Conditions>
            <Operator>Any</Operator>
          </CompoundFilterCondition>
        </Condition>
      </Filter>
    <Filter Type="String">
      <Mode>Cleared</Mode>
      <Operand>
```

```

        <QueryExpression xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
i:type="QueryFieldExpression">
            <Field>Color</Field>
            <Instance i:type="QdmEntityPlaceholderExpression">
                <Target>Sandbox.Media</Target>
            </Instance>
        </QueryExpression>
    </Operand>
</Filter>
</Filters>
</SectionFilters>
<ReportItems>
    <DataRegion Name="Tablix1">
        <Filters>
            <Filters
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2011/01/reportstate">
                <Filter Type="Decimal">
                    <Mode>Advanced</Mode>
                    <IsMeasure />
                    <Operand>
                        <QueryExpression xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
i:type="QueryCalculateExpression">
                            <Argument i:type="QueryFunctionExpression">
                                <FunctionName>Core.Sum</FunctionName>
                                <Arguments>
                                    <QueryExpression i:type="QueryProjectExpression">
                                        <Input>
                                            <Expression i:type="QueryScanExpression">
                                                <Target>Sandbox.Media</Target>
                                            </Expression>
                                            <VariableName>Media</VariableName>
                                        </Input>
                                        <Projection i:type="QueryFieldExpression">
                                            <Field>Gross</Field>
                                            <Instance i:type="QueryVariableReferenceExpression">
                                                <VariableName>Media</VariableName>
                                            </Instance>
                                        </Projection>
                                    </QueryExpression>
                                </Arguments>
                            </Argument>
                        </QueryExpression>
                    </Operand>
                    <Condition>
                        <CompoundFilterCondition xmlns:i="http://www.w3.org/2001/XMLSchema-
instance">
                            <Conditions>
                                <FilterCondition i:type="SimpleFilterCondition">
                                    <LeftExpression i:type="QueryCalculateExpression">
                                        <Argument i:type="QueryFunctionExpression">
                                            <FunctionName>Core.Sum</FunctionName>
                                            <Arguments>
                                                <QueryExpression i:type="QueryProjectExpression">
                                                    <Input>
                                                        <Expression i:type="QueryScanExpression">
                                                            <Target>Sandbox.Media</Target>
                                                        </Expression>
                                                        <VariableName>Media</VariableName>
                                                    </Input>

```

```

        <Projection i:type="QueryFieldExpression">
            <Field>Gross</Field>
            <Instance i:type="QueryVariableReferenceExpression">
                <VariableName>Media</VariableName>
            </Instance>
        </Projection>
    </QueryExpression>
</Arguments>
</Argument>
</LeftExpression>
<Operator>GreaterThanOrEqual</Operator>
<RightExpression i:type="QueryLiteralExpression">
    <Value xmlns:d5p1="http://www.w3.org/2001/XMLSchema"
i:type="d5p1:decimal">1</Value>
</RightExpression>
</FilterCondition>
<FilterCondition i:type="SimpleFilterCondition">
    <LeftExpression i:type="QueryCalculateExpression">
        <Argument i:type="QueryFunctionExpression">
            <FunctionName>Core.Sum</FunctionName>
            <Arguments>
                <QueryExpression i:type="QueryProjectExpression">
                    <Input>
                        <Expression i:type="QueryScanExpression">
                            <Target>Sandbox.Media</Target>
                        </Expression>
                        <VariableName>Media</VariableName>
                    </Input>
                    <Projection i:type="QueryFieldExpression">
                        <Field>Gross</Field>
                        <Instance i:type="QueryVariableReferenceExpression">
                            <VariableName>Media</VariableName>
                        </Instance>
                    </Projection>
                </QueryExpression>
            </Arguments>
        </Argument>
        </LeftExpression>
        <Operator>LessThan</Operator>
        <RightExpression i:type="QueryLiteralExpression">
            <Value xmlns:d5p1="http://www.w3.org/2001/XMLSchema"
i:type="d5p1:decimal">1000000000</Value>
        </RightExpression>
    </FilterCondition>
</Conditions>
</CompoundFilterCondition>
</Condition>
</Filter>
</Filters>
</DataRegion>
<DataRegion Name="Tablix3">
    <Filters>
        <Filters
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2011/01/reportstate">
            <Filter Type="String">
                <Mode>Basic</Mode>
                <Operand>

```



```
        </LeftExpression>
        <Operator>GreaterThanOrEqual</Operator>
        <RightExpression i:type="QueryLiteralExpression">
          <Value xmlns:d5p1="http://www.w3.org/2001/XMLSchema"
i:type="d5p1:long">8049</Value>
        </RightExpression>
      </FilterCondition>
    </Conditions>
  </CompoundFilterCondition>
</Condition>
</Filter>
</Filters>
</SectionFilters>
</ReportSection>
</ReportSections>
</ReportState>
```

## 4 Security

### 4.1 Security Considerations for Implementers

Because Report State documents only contain metadata that describe filters for a report definition, there are not security considerations for Report State documents themselves.

## 5 Appendix A: Full XML Schema

For ease of implementation, the following sections provide the full XML schemas for this protocol.

Schema name	Prefix	Section
Report State Schema	xs:	<a href="#">5.1</a>
Serialization Schema	xs:	<a href="#">5.2</a>

### 5.1 Report State Schema

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema
targetNamespace="http://schemas.microsoft.com/sqlserver/reporting/2011/01/reportstate"
elementFormDefault="qualified"
xmlns="http://schemas.microsoft.com/sqlserver/reporting/2011/01/reportstate"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
version="1.0">

  <xs:element name="ReportState">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="FilterAreaVisibility" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="Collapsed"/>
              <xs:enumeration value="Expanded"/>
              <xs:enumeration value="Hidden"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
        <xs:element name="Theme" minOccurs="0"/>
        <xs:element name="ReportSections" minOccurs="0" type="ReportSectionsType" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="ReportSectionsType">
    <xs:sequence>
      <xs:element name="ReportSection" type="ReportSectionType" minOccurs="0"
maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="ReportSectionType">
    <xs:sequence>
      <xs:element name="SectionFilters" type="SectionFiltersType" minOccurs="0" />
      <xs:element name="ReportItems" type="ReportItemsType" minOccurs="0" />
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="SectionFiltersType">
    <xs:sequence>
      <xs:element name="Filters" type="FiltersType" />
    </xs:sequence>
  </xs:complexType>
```

```

<xs:complexType name="FiltersType">
  <xs:sequence>
    <xs:element name="Filter" type="FilterType" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="FilterType">
  <xs:sequence>
    <xs:element name="Mode">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="Advanced" />
          <xs:enumeration value="Basic" />
          <xs:enumeration value="Cleared" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="IsMeasure" minOccurs="0" />
    <xs:element name="Operand">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="QueryExpression" type="QueryExpression"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="Condition" minOccurs="0">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="CompoundFilterCondition" type="CompoundFilterCondition"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="Type" type="PrimitiveType" />
</xs:complexType>

<xs:simpleType name="PrimitiveType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Boolean" />
    <xs:enumeration value="Byte" />
    <xs:enumeration value="DateTime" />
    <xs:enumeration value="Decimal" />
    <xs:enumeration value="Double" />
    <xs:enumeration value="Guid" />
    <xs:enumeration value="Single" />
    <xs:enumeration value="SByte" />
    <xs:enumeration value="Int16" />
    <xs:enumeration value="Int32" />
    <xs:enumeration value="Int64" />
    <xs:enumeration value="String" />
    <xs:enumeration value="Time" />
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="FilterCondition" abstract="true" />

<xs:complexType name="SimpleFilterCondition">
  <xs:complexContent>

```

```

<xs:extension base="FilterCondition">
  <xs:sequence>
    <xs:element name="LeftExpression" type="QueryExpression" />
    <xs:element name="Not" type="xs:boolean" minOccurs="0" />
    <xs:element name="Operator">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="Equal" />
          <xs:enumeration value="GreaterThan" />
          <xs:enumeration value="GreaterThanOrEqual" />
          <xs:enumeration value="LessThanOrEqual" />
          <xs:enumeration value="LessThan" />
          <xs:enumeration value="Contains" />
          <xs:enumeration value="StartsWith" />
          <xs:enumeration value="DateTimeEqualToSecond" />
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    <xs:element name="RightExpression" type="QueryExpression" minOccurs="0" />
  </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="CompoundFilterCondition">
  <xs:complexContent>
    <xs:extension base="FilterCondition">
      <xs:sequence>
        <xs:element name="Conditions" minOccurs="0">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="FilterCondition" type="FilterCondition" minOccurs="0"
maxOccurs="unbounded" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Operator" minOccurs="0">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="All" />
              <xs:enumeration value="Any" />
              <xs:enumeration value="NotAll" />
              <xs:enumeration value="NotAny" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="ReportItemsType">
  <xs:sequence>
    <xs:element name="DataRegion" type="DataRegionType" maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="DataRegionType">
  <xs:sequence>

```

```

    <xs:element name="Filters">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Filters" type="FiltersType" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="Name" type="xs:string" />
</xs:complexType>

<xs:complexType name="TypeUsage">
  <xs:sequence>
    <xs:element name="PrimitiveType" type="PrimitiveType" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="QueryExpressionBinding">
  <xs:sequence>
    <xs:element name="Expression" type="QueryExpression"/>
    <xs:element name="VariableName" type="xs:string" />
  </xs:sequence>
</xs:complexType>

<xs:complexType name="QueryExpressionList">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="QueryExpression" type="QueryExpression"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="QueryExpression" abstract="true" />

<xs:complexType name="QueryCalculateExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="Argument" type="QueryExpression" />
        <xs:element name="Filters" type="QueryExpressionList" minOccurs="0" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="QueryFieldExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="Field" type="xs:string" />
        <xs:element name="Instance" type="QueryExpression" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="QueryFunctionExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="FunctionName">

```

```

        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:enumeration value="Core.Sum" />
                <xs:enumeration value="Core.Count" />
                <xs:enumeration value="Core.DistinctCount" />
                <xs:enumeration value="Core.Average" />
                <xs:enumeration value="Core.Min" />
                <xs:enumeration value="Core.Max" />
            </xs:restriction>
        </xs:simpleType>
    </xs:element>
    <xs:element name="Arguments" type="QueryExpressionList" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:complexType name="QueryLiteralExpression">
    <xs:complexContent>
        <xs:extension base="QueryExpression">
            <xs:sequence>
                <xs:element name="Value" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="QueryMeasureExpression">
    <xs:complexContent>
        <xs:extension base="QueryExpression">
            <xs:sequence>
                <xs:element name="Measure" type="xs:string" />
                <xs:element name="Target" type="xs:string" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="QueryNullExpression">
    <xs:complexContent>
        <xs:extension base="QueryExpression">
            <xs:sequence>
                <xs:element name="Type" type="TypeUsage" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="QueryProjectExpression">
    <xs:complexContent>
        <xs:extension base="QueryExpression">
            <xs:sequence>
                <xs:element name="Input" type="QueryExpressionBinding" />
                <xs:element name="Projection" type="QueryExpression" />
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

```

```

<xs:complexType name="QueryScanExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="Target" type="xs:string" />
        <xs:element name="ExcludeBlankRow" type="xs:boolean" minOccurs="0" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="QueryVariableReferenceExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="VariableName" type="xs:string" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="QdmEntityPlaceholderExpression">
  <xs:complexContent>
    <xs:extension base="QueryExpression">
      <xs:sequence>
        <xs:element name="Target" type="xs:string" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:element name="ReportSlicerState">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="CompoundFilterCondition" type="CompoundFilterCondition"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>

```

## 5.2 Serialization Schema

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="qualified"
  elementFormDefault="qualified"
  targetNamespace =
    "http://schemas.microsoft.com/2003/10/Serialization/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="http://schemas.microsoft.com/2003/10/Serialization/">

  <!-- Top-level elements for primitive types. -->
  <xs:element name="anyType" nillable="true" type="xs:anyType"/>
  <xs:element name="anyURI" nillable="true" type="xs:anyURI"/>
  <xs:element name="base64Binary"
    nillable="true" type="xs:base64Binary"/>

```

```

<xs:element name="boolean" nillable="true" type="xs:boolean"/>
<xs:element name="byte" nillable="true" type="xs:byte"/>
<xs:element name="dateTime" nillable="true" type="xs:dateTime"/>
<xs:element name="decimal" nillable="true" type="xs:decimal"/>
<xs:element name="double" nillable="true" type="xs:double"/>
<xs:element name="float" nillable="true" type="xs:float"/>
<xs:element name="int" nillable="true" type="xs:int"/>
<xs:element name="long" nillable="true" type="xs:long"/>
<xs:element name="QName" nillable="true" type="xs:QName"/>
<xs:element name="short" nillable="true" type="xs:short"/>
<xs:element name="string" nillable="true" type="xs:string"/>
<xs:element name="unsignedByte" nillable="true" type="xs:unsignedByte"/>
<xs:element name="unsignedInt" nillable="true" type="xs:unsignedInt"/>
<xs:element name="unsignedLong" nillable="true" type="xs:unsignedLong"/>
<xs:element name="unsignedShort" nillable="true" type="xs:unsignedShort"/>

<!-- Primitive types introduced for certain .NET simple types. -->
<xs:element name="char" nillable="true" type="tns:char"/>
<xs:simpleType name="char">
  <xs:restriction base="xs:int"/>
</xs:simpleType>

<!-- xs:duration is restricted to an ordered value space,
to map to System.TimeSpan -->
<xs:element name="duration" nillable="true" type="tns:duration"/>
<xs:simpleType name="duration">
  <xs:restriction base="xs:duration">
    <xs:pattern
      value="\-?P(\d*D)?(T(\d*H)?(\d*M)?(\d*(\.\d*)?S)?)?" />
    <xs:minInclusive value="-P10675199DT2H48M5.4775808S"/>
    <xs:maxInclusive value="P10675199DT2H48M5.4775807S"/>
  </xs:restriction>
</xs:simpleType>

<xs:element name="guid" nillable="true" type="tns:guid"/>
<xs:simpleType name="guid">
  <xs:restriction base="xs:string">
    <xs:pattern value="[\da-fA-F]{8}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{4}-[\da-fA-F]{12}" />
  </xs:restriction>
</xs:simpleType>

<!-- This is used for schemas exported from ISerializable type. -->
<xs:attribute name="FactoryType" type="xs:QName"/>
</xs:schema>

```

## 6 Appendix C: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft® SQL Server® 2012

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product 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.

## 7 Change Tracking

This section identifies changes that were made to the [MS-RDLRS] protocol document between the January 2012 and February 2012 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

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 or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

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 **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated**.

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact [protocol@microsoft.com](mailto:protocol@microsoft.com).

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
<a href="#">2.2 ReportState</a>	866762 Added description of the Theme element.	Y	New content added.
<a href="#">3.1 Report State</a>	866762 Added Theme element to the XSD.	Y	New content added.
<a href="#">5.1 Report State Schema</a>	866762 Added Theme element to the XSD.	Y	New content added.

## 8 Index

### C

[Change tracking](#) 40

### T

[Tracking changes](#) 40

### U

[updates to this document](#) 40