[MS-DPRDL-Diff]:

Report Definition Language Data Portability Overview

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 as well as overviews of the interaction among each of these technologiessupport. 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 maycan make copies of it in order to develop implementations of the technologies that are described in the Open Specifications this documentation and maycan distribute portions of it in your implementations usingthat use these technologies or in your documentation as necessary to properly document the implementation. You maycan also distribute in your implementation, with or without modification, any schema, IDL'sschemas, 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 maymight cover your implementations of the technologies described in the Open Specifications. documentation. Neither this notice nor Microsoft's delivery of thethis documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specification maySpecifications 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 the Open Specificationsthis 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 maymight 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, e-mailemail 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 dodocumentation 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 standardstandards specifications and network programming art, and assumes, 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
6/4/2010	0.1	Major	First release.
9/3/2010	0.1.1	Editorial	Changed language and formatting in the technical content.
2/9/2011	0.1.1	None	No changes to the meaning, language, or formatting of the technical content.
7/7/2011	1.0	Major	Updated and revised the technical content.
11/3/2011	1.0	Major	Updated and revised the technical content.
1/19/2012	1.0	None	No changes to the meaning, language, or formatting of the technical content.
2/23/2012	1.0	None	No changes to the meaning, language, or formatting of the technical content.
3/27/2012	1.0	None	No changes to the meaning, language, or formatting of the technical content.
5/24/2012	1.0	None	No changes to the meaning, language, or formatting of the technical content.
6/29/2012	1.0	None	No changes to the meaning, language, or formatting of the technical content.
7/16/2012	2.0	Major	Updated and revised the technical content.
10/8/2012	2.0	None	No changes to the meaning, language, or formatting of the technical content.
10/23/2012	2.0	None	No changes to the meaning, language, or formatting of the technical content.
3/26/2013	2.1	Minor	Clarified the meaning of the technical content.
6/11/2013	2.1	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	2.1	None	No changes to the meaning, language, or formatting of the technical content.
12/5/2013	2.1	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2014	2.1	None	No changes to the meaning, language, or formatting of the technical content.
5/20/2014	2.1	None	No changes to the meaning, language, or formatting of the technical content.
5/10/2016	2.1	None	No changes to the meaning, language, or formatting of the technical content.
8/16/2017	2.1	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1   Introduction

The Report Definition Language Data Portability Overview document provides an overview of data portability for the Microsoft SQL Server Reporting Services Report Definition Language (RDL) system. This system includes a repository for RDL documents, such as the Reporting Services report server catalog, SharePoint products and technologies, or the file system and RDL documents.

RDL documents [MS-RDL] represent the definition of the reports. These documents are either set or retrieved in the report server catalog by using the SOAP endpoints—ReportService2005 [MS-RSWSRMNM2005], ReportService2006 [MS-RSWSRMSM2006], or ReportService2010 [MS-RSWSRM2010]—or they are opened or saved in a SharePoint library or in the file system.

1.1   Glossary

This document uses the following terms:

data source: A physical data source.

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.

Entity Data Model (EDM): A set of concepts that describes the structure of data, regardless of its stored form.

Open Packaging Conventions (OPC): An open standard for a portable container technology that defines a structured way to store application data with related resources by using a standard .ZIP file format. OPC is a component of Office Open XML File Formats [ECMA-376].

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.

SOAP: A lightweight protocol for exchanging structured information in a decentralized, distributed environment. SOAP uses XML technologies to define an extensible messaging framework, which provides a message construct that can be exchanged over a variety of underlying protocols. The framework has been designed to be independent of any particular programming model and other implementation-specific semantics. SOAP 1.2 supersedes SOAP 1.1. See [SOAP1.2-1/2003].

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.

[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

[ECMA-376-3/2] ECMA, "Information technology – Document description and processing languages – Office Open XML File Formats – Part 3: Markup Compatibility and Extensibility", 2nd edition, Standard ECMA-376-3, December 2008, http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-376,%20Second%20Edition,%20Part%203%20-%20Markup%20Compatibility%20and%20Extensibility.zip

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

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

[MS-RDL] Microsoft Corporation, "Report Definition Language File Format".

[MS-RSWSRM2010] Microsoft Corporation, "Report Server Web Service for Report Management: ReportService2010".

[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".

[MS-SSAS] Microsoft Corporation, "SQL Server Analysis Services Protocol".

[MSDN-RMADS] Microsoft Corporation, "Retrieving Metadata from an Analytical Data Source","", http://msdn.microsoft.com/en-us/library/ms123485.aspx

2 Data Portability Scenarios

2.1 Third-Party Reporting Platform Consuming RDL Documents in the Report Server Database

2.1.1 Data Description

The RDL [MS-RDL] document contains the definition of a report , with information about how to connect to data sources, which fields are used from the datasets that are retrieved from the data sources, how the data is aggregated, and the structure and layout of the report.

This RDL data is used to process data and to render a report. The data is stored in the report server database when Reporting Services is running in native mode.

This RDL data is created by using a Reporting Services RDL authoring tool (Report Builder or Report Designer in the Business Intelligence Development Studio), by using a third-party RDL authoring tool, or by using a text editor.

2.1.2 Format and Protocol Summary

The following table provides a comprehensive list of the formats and protocols that are used in this scenario.

Protocol or format name	Description	Reference
ReportService2005 web service protocol	This protocol is used to communicate with the report server to execute report server database operations. The ReportService2005 web service protocol is available in Microsoft SQL Server 2005, Microsoft SQL Server 2008, Microsoft SQL Server 2008 R2, and Microsoft SQL Server 2012.	[MS-RSWSRMNM2005]
ReportService2010 web service protocol	This protocol is used to communicate with the report server to execute report server database operations. The ReportService2010 web service protocol is available in SQL Server 2008 R2 and SQL Server 2012.	[MS-RSWSRM2010]
Report Definition Language file format	This format specifies the file format for SQL Server Report Definition Language, a file type that is used to represent the metadata for defining a report.	[MS-RDL]

2.1.3 Data Portability Methodology

For this scenario, the documents that contain the RDL data are extracted from the report server database one by one and stored in a file on the file system. The method of extracting the RDL data from the report server database for use in a third-party reporting platform in this scenario is to use the SOAP endpoints that are provided by the report server.

In this scenario, the ReportService2005 [MS-RSWSRMNM2005] and the ReportService2010 [MS-RSWSRM2010] SOAP endpoints enable implementers to programmatically extract the data from the report server database.

To extract the data, follow these steps:

1. Create a folder on the client machine for storing the retrieved RDL documents.

2. Use a SOAP proxy to access the ReportService2005 web service or the ReportService2010 web service, and then obtain the list of RDL documents in the report server database by using the **ListChildren**() SOAP web method.

    1. For the first call to **ListChildren**(), use "/" as the value for the *Item* parameter. This returns each **CatalogItem** that is at the root level.

    2. For each **CatalogItem** returned, follow these steps:

        1. If the **CatalogItem** is of type **Report**, store the **CatalogItem.Path**.

        2. If the **CatalogItem** is of type **Folder**, call **ListChildren** with **CatalogItem.Path** as the value for the *Item* parameter and go to substep 2.

3. Retrieve each RDL definition from the report server database. For each item path that is stored in step 2, follow these steps:

    1. Call the **GetReportDefinition**() SOAP web method, passing the item path as the value for the *Report* parameter.

    2. Create a file in the folder that was created in main step 1 of this procedure. Use the returned byte array as the contents of the file.

4. Use the RDL documentation [MS-RDL] to interpret the RDL data that was retrieved in the previous step for use in the third-party reporting platform.

2.1.3.1 Preconditions

Ensure that the Reporting Services service is started on the server. Grant the appropriate permissions to the user who is using the ReportService2005 or ReportService2010 SOAP endpoint to access the report server database.

2.1.3.2 Versioning

None.

2.1.3.3 Error Handling

None.

2.1.3.4 Coherency Requirements

There are no special coherency requirements.

2.1.3.5 Additional Considerations

There are no additional considerations.

2.2   Third-Party Reporting Platform Consuming RDL Documents in SharePoint

2.2.1 Data Description

The Report Definition Language (RDL) [MS-RDL] document contains the definition of a report with information about how to connect to data sources, which fields are used from the datasets retrieved from the data sources, how the data is aggregated, and the structure and layout of the report.

This RDL data is used to process data and to render a report. The data is stored in both the report server database and the SharePoint repository when running Reporting Services in SharePoint integrated mode.

This data is created by using a Reporting Services RDL authoring tool (Report Builder or Report Designer in the Business Intelligence Development Studio), by using a third-party RDL authoring tool, or by using a text editor.

In addition to an RDL document, there is an RDLX file. An RDLX file is a package that is compatible with a compressed (.zip) file. The contents of an RDLX file can be viewed by renaming it to a .zip file and then opening it with any file compression program that creates .zip files. The RDLX file contains an RDL [MS-RDL] document, an optional Report State [MS-RDLRS] document, and an optional Extended Properties document. (Documentation on the Extended Properties document is provided throughout this overview document.) An RDLX file might also contain optional embedded image files that can be used in the report and in other files that are for cache purposes only.

The structure within the RDLX file follows the rules that are outlined in the Open Packaging Conventions (OPC) [ECMA-376-2/2]. Therefore, a user can extract the content of an RDLX file by implementing the OPC conventions, either manually or through a code library such as the System.IO.Packaging library that is included in the .NET Framework [MSDN-RMADS].

The following table lists the OPC relationship types that are required to extract content from an RDLX file.

Document	OPC relationship type
RDL	http://schemas.microsoft.com/sqlserver/reporting/2011/01/reportpackage/relationships/reportdefinition
Report State	http://schemas.microsoft.com/sqlserver/reporting/2011/01/reportpackage/relationships/reportstate
Extended Properties	http://schemas.microsoft.com/sqlserver/reporting/2012/01/reportpackage/relationships/extendedproperties

The RDL document can be extracted by using the top-level RDL relationship and can then be viewed in a text editor, in the same manner that other RDL documents are viewed.

The Report State document can be extracted by using the RDL-level Report State relationship and can then be viewed in a text editor. The state information in the Report State document makes references to an Entity Data Model (EDM) that has to be extracted from SQL Server Analysis Services by using the methodology that is described in Data Portability Methodology (section 2.2.3).

The Extended Properties document can be extracted by using the top-level Extended Properties relationship and can then be viewed in a text editor. The Extended Properties document supplies a set of metadata properties for **RDLXReport**. For instance, one of those metadata properties maycan be used to version **RDLXReport**. A versional **RDLXReport** allows a client application to programmatically determine if it can process **RDLXReport** or not.

The RDLX document is created by using a Reporting Services report authoring tool named Microsoft Power View or by using a third-party RDL authoring tool.

2.2.2 Format and Protocol Summary

The following table provides a comprehensive list of the formats and protocols that are used in this scenario.

Protocol or format name	Description	Reference
ReportService2006 web service protocol	This protocol is used to communicate with the report server in SharePoint integrated mode to execute report server database operations.	[MS-RSWSRMSM2006]
Report Definition Language file format	This format is used to specify the file format for SQL Server Report Definition Language (RDL), a file type that is used to represent the metadata for defining a report.	[MS-RDL]
Report State file format	This format is used to specify the file format for SQL Server Report State, a file type that is used to represent filter state of a report.	[MS-RDLRS]
Extended Properties file format	This format is used to specify the file format for RDLX Extended Properties, a file type that is used to represent the metadata properties for the RDLX file.	For more information on the Extended Properties document and format, see sections 2.2.1, 2.2.3, 2.2.3.5, and 3.
Conceptual Schema Definition Language (CSDL)	This format is used to specify the EDM that is used by both the RDL and Report State formats.	[MC-CSDL]

2.2.3 Data Portability Methodology

Because the data is stored in both the SharePoint repository and the report server database, the approach for this scenario is to extract the data by accessing the report server database by using the steps outlined in section 2.1.3. However, instead of using the ReportService2005 [MS-RSWSRMNM2005] SOAP endpoint, the ReportService2006 [MS-RSWSRMSM2006] SOAP endpoint is used in this scenario.

The ReportService2006 SOAP endpoint enables implementers to programmatically manage objects on a report server that is configured for SharePoint integrated mode.

To extract the data, follow these steps:

1. Create a folder on the client machine for storing the retrieved RDL documents.

2. Use a SOAP proxy to access the ReportService2006 web service or the ReportService2010 web service, and then obtain the list of RDL documents in the report server database by using the **ListChildren**() SOAP web method.

    1. For the first call to **ListChildren**(), use "/" as the value for the *Item* parameter. This returns each **CatalogItem** that is at the root level.

    2. For each **CatalogItem** returned, follow these steps:

        1. If the **CatalogItem** is of type **Report** or **RDLXReport**, store the **CatalogItem.Path**.

        2. If the **CatalogItem** is of type **Folder**, call **ListChildren** with **CatalogItem.Path** as the value for the *Item* parameter and go to substep 2.

3. Retrieve each RDL definition from the report server database. For each item path that is stored in step 2, follow these steps:

1. For an item of type **Report**, call the **GetReportDefinition**() SOAP web method, passing the item path as the value for the *Report* parameter. For an item of type **RDLXReport**, call the **GetItemDefinition**() SOAP web method, passing the item path as the value for the *Item* parameter.

2. Create a file in the folder that was created in main step 1 of this procedure. Use the returned byte array as the contents of the file.

4. For an item of type **RDLXReport**, rename the file to a .zip file and open it with any .zip tool.

   1. Extract the .rels file and open it by using a text editor.

   2. Find the RDL relationship, and then use its **Target** attribute value to obtain the location of the RDL file.

   3. Extract the RDL file from this location within the .zip file.

   4. To extract the optional Report State file, append ".rels" to the location of the Report RDL file, and then extract this .rels file from the .zip file and open it by using a text editor.

   5. Find the Report State relationship, and then use its **Target** attribute value to obtain the location of the Report State file.

   6. Extract the Report State file from this location within the .zip file.

   7. To extract the Extended Properties file, find the Extended Properties relationship in the .rels file that was extracted in step 1, and then use its Target attribute value to obtain the location of the Extended Properties file.

   8. Extract the Extended Properties file from this location within the .zip file.

5. Use the RDL documentation [MS-RDL] to interpret the RDL data that was retrieved in step 4 for use in the third-party reporting platform.

6. Use the Report State documentation [MS-RDLRS] to interpret the Report State that was retrieved in step 4 for use in the third-party reporting platform.

7. Use the Extended Properties documentation (see the Extended Properties document and format information in this document) to interpret the Extended Properties data that was retrieved in step 4 for use in the third-party reporting platform.

To extract the EDM schema [MC-CSDL], follow these steps:

1. In the RDL data, find either the **ConnectString** element (Embedded Data Source) or the **DataSourceReference** element (Shared DataSource), and then extract its value.

2. If the value came from a **DataSourceReference** element, use a SOAP proxy to access the ReportService2010 web service, and then call the **GetDataSourceContents** () SOAP web method with this value. The **GetDataSourceContents** method returns a **DataSourceDefinition** value that contains a **ConnectString** property.

3. Create an instance of a .NET ADOMD connection by using the **ConnectString** value, and then call the **GetSchemaDataset** method with the schema name DISCOVER_CSDL_METADATA (see [MS-SSAS] section 3.1.4.2.2.1.3.61). Use the [MS-RMADS] documentation for more information about how to do this.

4. Use the DISCOVER_CSDL_METADATA documentation to interpret the data that was retrieved in the previous step.

5. After the CSDL is extracted, use the CSDL documentation to interpret it.

2.2.3.1 Preconditions

Ensure that the Reporting Services service is started on the server and that the SharePoint service is running. Grant the appropriate permissions to the user who is using the ReportService2006 SOAP endpoint to access the report server database.

2.2.3.2 Versioning

None.

2.2.3.3 Error Handling

None.

2.2.3.4 Coherency Requirements

There are no special coherency requirements.

2.2.3.5 Additional Considerations

The Extended Properties document ~~may~~can contain an optional **MustUnderstand** attribute on the top-level **Properties** element.

Attribute	Namespace	Reference
MustUnderstand	http://schemas.openxmlformats.org/markup-compatibility/2006	[ECMA-376-3/2]

The [ECMA-376-3/2] documentation can be used to interpret the **MustUnderstand** attribute.

2.3   Third-Party Reporting Platform Consuming RDL Documents in the File System

2.3.1   Data Description

The Report Definition Language (RDL) [MS-RDL] document contains the definition of a report , with information about how to connect to data sources, which fields are used from the datasets retrieved from the data sources, how the data is aggregated, and the structure and layout of the report. This RDL data is used to process data and to render a report. The data is stored in the file system on the local computer.

This data is created by using a Reporting Services RDL authoring tool (Report Builder or Report Designer in the Business Intelligence Development Studio), by using a third-party RDL authoring tool, or by using a text editor.

2.3.2   Format and Protocol Summary

No formats or protocols are used in this scenario.

2.3.3   Data Portability Methodology

In this scenario, the RDL data is stored in the file system as reports (*.rdl files). By default, reports are saved in the Documents folder on the local machine (in Windows Vista, this folder is C:\Users\<username>\Documents\). Use the RDL documentation [MS-RDL] to interpret the RDL data in these files.

2.3.3.1 Preconditions

None.

2.3.3.2 Versioning

None.

2.3.3.3 Error Handling

None.

2.3.3.4 Coherency Requirements

There are no special coherency requirements.

2.3.3.5 Additional Considerations

There are no additional considerations.

3   Appendix A: Full XML Schema for the Extended Properties Format

For ease of implementation, the following full XML schema is provided here.

```xml
<?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (c) Microsoft Corporation.  All rights reserved. -->
<xs:schema
targetNamespace="http://schemas.microsoft.com/sqlserver/reporting/2012/01/extendedproperties"
    elementFormDefault="qualified"
    xmlns="http://schemas.microsoft.com/sqlserver/reporting/2012/01/extendedproperties"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    version="1.0">
  <xs:element name="Properties">
    <xs:complexType>
      <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:any namespace="##other" processContents="lax" />
      </xs:choice>
      <xs:anyAttribute namespace="##other" processContents="lax" />
    </xs:complexType>
  </xs:element>

</xs:schema>
```

4   Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

5   Index

C

Change tracking 14

F

Full XML schema 13

G

Glossary 4

I

Informative references 4
Introduction 4

R

References 4

T

Tracking changes 14

X

XML schema 13