

# [MS-DPRDL]: Report Definition Language Data Portability Overview

---

## 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
06/04/2010	0.1	Major	First release.
09/03/2010	0.1.1	Editorial	Changed language and formatting in the technical content.
02/09/2011	0.1.1	No change	No changes to the meaning, language, or formatting of the technical content.
07/07/2011	1.0	Major	Significantly changed the technical content.
11/03/2011	1.0	Major	Significantly changed the technical content.

# Contents

<b>1 Introduction</b> .....	<b>4</b>
1.1 Glossary .....	4
1.2 References.....	4
<b>2 Data Portability Scenarios</b> .....	<b>6</b>
2.1 Third-Party Reporting Platform Consuming RDL Documents in the Report Server Database .....	6
2.1.1 Data Description .....	6
2.1.2 Format and Protocol Summary .....	6
2.1.3 Data Portability Methodology .....	6
2.1.3.1 Preconditions .....	7
2.1.3.2 Versioning .....	7
2.1.3.3 Error Handling.....	7
2.1.3.4 Coherency Requirements.....	7
2.1.3.5 Additional Considerations.....	7
2.2 Third-Party Reporting Platform Consuming RDL Documents in SharePoint.....	7
2.2.1 Data Description .....	7
2.2.2 Format and Protocol Summary .....	8
2.2.3 Data Portability Methodology .....	9
2.2.3.1 Preconditions .....	10
2.2.3.2 Versioning .....	10
2.2.3.3 Error Handling.....	10
2.2.3.4 Coherency Requirements.....	10
2.2.3.5 Additional Considerations.....	11
2.3 Third-Party Reporting Platform Consuming RDL Documents in the File System .....	11
2.3.1 Data Description .....	11
2.3.2 Format and Protocol Summary.....	11
2.3.3 Data Portability Methodology .....	11
2.3.3.1 Preconditions .....	11
2.3.3.2 Versioning .....	11
2.3.3.3 Error Handling.....	11
2.3.3.4 Coherency Requirements.....	11
2.3.3.5 Additional Considerations.....	11
<b>3 Change Tracking</b> .....	<b>12</b>
<b>4 Index</b> .....	<b>14</b>

# 1 Introduction

The Microsoft® SQL Server® Reporting Services system includes a repository for Report Definition Language (RDL) documents, such as the Reporting Services report server database, Microsoft SharePoint Products and Technologies, or the file system and RDL documents [\[MS-RDL\]](#).

RDL documents represent the definition of the **report**. These documents are either set or retrieved in the report server database 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

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

**dataset**  
**report**

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

**Entity Data Model (EDM)**

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

**data source**

The following protocol abbreviations are used in this document:

**SOAP:** Simple Object Access Protocol

## 1.2 References

[ECMA-376-2/2] ECMA, "Office Open XML File Formats – Part 2", 2nd edition, ECMA-376, December 2008, <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-376,%20Second%20Edition,%20Part%202%20-%20Open%20Packaging%20Conventions.zip>

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

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

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

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

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

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

[MS-SSAS] Microsoft Corporation, "[SQL Server Analysis Services Protocol Specification](#)".

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

[MSDN-SIOPN] Microsoft Corporation, "System.IO.Packaging Namespace",  
<http://msdn.microsoft.com/en-us/library/system.io.packaging.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 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, and Microsoft® SQL Server® 2008 R2.	<a href="#">[MS-RWSRMNM2005]</a>
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.	<a href="#">[MS-RWSRM2010]</a>
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.	<a href="#">[MS-RDL]</a>

#### 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 [\[MS-RWSRMNM2005\]](#) and the ReportService2010 [\[MS-RWSRM2010\]](#) 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**, repeat steps 2.1 and 2.2.
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 in the item path as the value for the *Report* parameter.
  2. Create a file in the folder that was created in step 1. 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 [\[MS-RDL\]](#) 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 opening it with any file compression program that creates .zip files. The RDLX file contains an RDL [MS-RDL] document and an optional Report State [MS-RDLRS] document.

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	<a href="http://schemas.microsoft.com/sqlserver/reporting/2011/01/reportpackage/relationships/reportdefinition">http://schemas.microsoft.com/sqlserver/reporting/2011/01/reportpackage/relationships/reportdefinition</a>
Report State	<a href="http://schemas.microsoft.com/sqlserver/reporting/2011/01/reportpackage/relationships/reportstate">http://schemas.microsoft.com/sqlserver/reporting/2011/01/reportpackage/relationships/reportstate</a>

The RDL document can be extracted by using the top-level RDL relationship and can then be viewed in a text editor, just as 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, later in this document.

This 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.	<a href="#">[MS-RSWSRMSM2006]</a>
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	<a href="#">[MS-RDL]</a>



Protocol or format name	Description	Reference
	report.	
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.	<a href="#">[MS-RDLRS]</a>
Conceptual Schema Definition Language (CSDL)	This format is used to specify the <a href="#">EDM</a> that is used by both the RDL and Report State formats.	<a href="#">[MC-CSDL]</a>

### 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**, repeat steps 2.1 and 2.2.
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 in the item path as the value for the *Report* parameter. For an item of type **RDLXReport**, call the **GetItemDefinition()** SOAP Web method, passing in the item path as the value for the *Item* parameter.
  2. Create a file in the folder that was created in step 1. 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 from this location within the .zip file.
5. 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.
6. Use the Report State documentation [\[MS-RDLRS\]](#) to interpret the Report State that was retrieved in the previous step 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 **ConnectionString** 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, in turn, contains a **ConnectionString** property.
3. Create an instance of a .NET ADOMD connection by using the **ConnectionString** 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**

There are no additional considerations.

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

### **2.3.1 Data Description**

The 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 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\\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 Change Tracking

This section identifies changes that were made to the [MS-DPRDL] protocol document between the July 2011 and November 2011 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).

<b>Section</b>	<b>Tracking number (if applicable) and description</b>	<b>Major change (Y or N)</b>	<b>Change type</b>
<a href="#">2.2.3 Data Portability Methodology</a>	817041 Updated procedures to include changes made in SQL Server code-named "Denali" RC0.	Y	Content updated.

## 4 Index

### C

[Change tracking](#) 12

### T

[Tracking changes](#) 12