

[MS-DPSSAS]: SQL Server Analysis Services 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.
- **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	No change	No changes to the meaning, language, or formatting of the technical content.
02/09/2011	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
07/07/2011	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
11/03/2011	0.1	No change	No changes to the meaning, language, or formatting of the technical content.
01/19/2012	1.1	Minor	Clarified the meaning of the technical content.

Contents

1 Introduction	4
1.1 Glossary	4
1.2 References.....	4
2 Data Portability Scenarios	6
2.1 Exporting Metadata.....	6
2.1.1 Data Description	6
2.1.1.1 Unified Dimensional Model	6
2.1.1.2 Data Mining	7
2.1.2 Format and Protocol Summary.....	7
2.1.3 Data Portability Methodology	7
2.1.3.1 Using Microsoft SQL Server Management Studio	7
2.1.3.2 Using Analysis Management Objects	8
2.1.3.3 Preconditions	8
2.1.3.4 Versioning	8
2.1.3.5 Error Handling.....	8
2.1.3.6 Coherency Requirements	8
2.1.3.7 Additional Considerations.....	8
2.1.3.7.1 Data Source Connection String	9
2.2 Exporting Writeback Data	9
2.2.1 Data Description	9
2.2.1.1 Dimension Writeback	9
2.2.1.2 Cube and Partition Writeback	9
2.2.2 Format and Protocol Summary.....	9
2.2.3 Data Portability Methodology	10
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.....	10
3 Change Tracking	11
4 Index	13

1 Introduction

This document provides an overview of data portability scenarios between Microsoft SQL Server Analysis Services and a vendor's application.

Analysis Services provides a business intelligence (BI) platform that enables end users and IT professionals to efficiently analyze business data. Two primary workloads exist for achieving this:

- **Corporate BI:** In this mode, IT professionals use Business Intelligence Development Studio and Microsoft SQL Server Management Studio to build and manage Analysis Services installations.
- **Self-Service BI:** In this mode, end users build their own solutions by using PowerPivot technologies.

In either mode, Analysis Services databases are built and used by the client tools. Unless specified otherwise, concepts and mechanisms described in this document are applicable to both workloads.

1.1 Glossary

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

cube

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

**analysis server
dimension
measure
partition**

1.2 References

[MS-OFCGLOS] Microsoft Corporation, "[Microsoft Office Master Glossary](#)".

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

[MSDN-AMO] Microsoft Corporation, "Analysis Management Objects",
<http://msdn.microsoft.com/en-us/library/ms124924.aspx>

[MSDN-BIDS] Microsoft Corporation, "Introducing Business Intelligence Development Studio",
<http://msdn.microsoft.com/en-us/library/ms173767.aspx>

[MSDN-PROC] Microsoft Corporation, "Analysis Services 2005 Processing Architecture",
[http://msdn.microsoft.com/en-us/library/ms345142\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms345142(SQL.90).aspx)

[MSDN-SSMS] Microsoft Corporation, "Introducing SQL Server Management Studio",
<http://msdn.microsoft.com/en-us/library/ms174173.aspx>

[MSDN-UDM] Microsoft Corporation, "Unified Dimensional Model", [http://msdn.microsoft.com/en-us/library/ms174783\(SQL.90\).aspx](http://msdn.microsoft.com/en-us/library/ms174783(SQL.90).aspx)

[MSFT-DM] Microsoft Corporation, "Data Mining Concepts (Analysis Services – Data Mining)",
<http://technet.microsoft.com/en-us/library/ms174949.aspx>

[MSFT-WBDIM] Microsoft Corporation, "Write-Enabled Dimensions",
<http://technet.microsoft.com/en-us/library/ms174540.aspx>

[MSFT-WBPT] Microsoft Corporation, "Write-Enabled Partitions", <http://technet.microsoft.com/en-us/library/ms174750.aspx>

2 Data Portability Scenarios

Analysis Services imports data from a variety of data sources and makes this data available for analysis by end users. Analysis Services architecture is designed with the assumption that the underlying data source is the master store of this data. This assumption holds for data pushed to Analysis Services through a mechanism called push-mode processing. For more information about push-mode processing, see [\[MSDN-PROC\]](#). As such, Analysis Services does not provide an efficient bulk data export utility and instead depends on the capabilities of the underlying data source for this purpose.

At the same time, Analysis Services provides support for exporting the definition of objects defined by IT professionals. In addition, Analysis Services enables end users to write data back into Analysis Services. This section describes how to export this data.

2.1 Exporting Metadata

Third-party applications can export definitions of user-created objects stored within Analysis Services.

2.1.1 Data Description

2.1.1.1 Unified Dimensional Model

The Analysis Services metadata model, called a Unified Dimensional Model (UDM), provides a bridge between users and the data sources. A UDM is constructed over one or more physical data sources, and it allows end-user queries using one of a variety of client tools, such as Microsoft® Office Excel®.

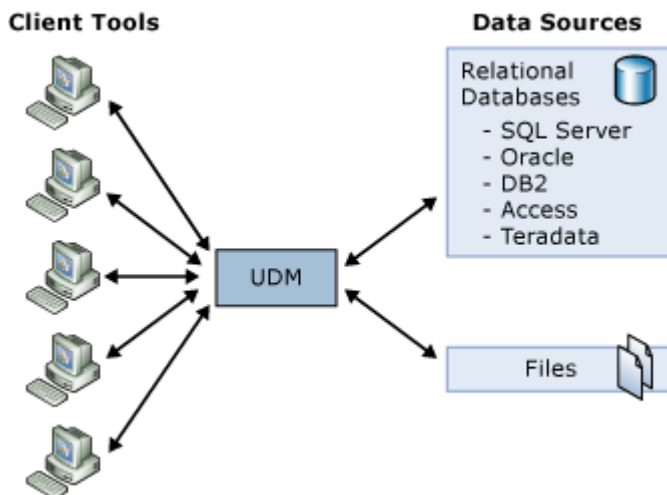


Figure 1: Unified Dimensional Model

A UDM contains information about:

- Data source connections.
- A schema snapshot for data that exists in a data source.
- The user-visible concepts, such as **dimensions**, hierarchies, and key performance indicators.

- Mapping between the user-visible concepts and the underlying data sources.
- Calculations that encapsulate business logic, such as a three-month moving average.
- Security roles and associated authorizations.

For more information about the UDM, see [\[MSDN-UDM\]](#).

A UDM is typically defined by IT professionals using Business Intelligence Development Studio. During the development process, UDM metadata is stored in proprietary XML-based files. Once this UDM definition is complete, it can be deployed by Business Intelligence Development Studio to an **analysis server** by using the SQL Server Analysis Services Protocol [\[MS-SSAS\]](#) where it is stored in a proprietary format. For more information about Business Intelligence Development Studio, see [\[MSDN-BIDS\]](#).

2.1.1.2 Data Mining

Data mining is the process of discovering actionable information from data by using various mathematical analysis techniques. Analysis Services provides data mining support. Within Analysis Services, data mining information is specified as part of a database. For more information about data mining, see [\[MSFT-DM\]](#).

Client tools use the SQL Server Analysis Services Protocol [\[MS-SSAS\]](#) for communicating with both UDM and data mining on an analysis server.

2.1.2 Format and Protocol Summary

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

Protocol or format name	Description	Reference
SQL Server Analysis Services Protocol	Specifies methods for a client to communicate with, and perform operations on, an analysis server.	[MS-SSAS]

2.1.3 Data Portability Methodology

2.1.3.1 Using Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studio allows implementers to manage instances of Analysis Service servers. For more information about SQL Server Management Studio, see [\[MSDN-SSMS\]](#).

To extract metadata by using SQL Server Management Studio, follow these steps:

1. Connect to Analysis Services.
 - For Analysis Services servers in multidimensional mode, please connect to the database by providing the server name or servername\instancename.
 - For PowerPivot workbooks, upload the PowerPivot workbook to a Microsoft PowerPivot for SharePoint server. Connect to the database by providing the URL of the PowerPivot workbook on the Microsoft SharePoint server.

- For Tabular projects in Business Intelligence Development Studio, deploy the Tabular Project to Analysis Services running in Tabular mode. Connect to the database by providing the server name or servername\instance name.
2. Select the database to be scripted. In the case of PowerPivot workbooks, you only have a single database.
 3. Right-click the database object to see the context menu, and then select **Script Database as**.
 4. Retrieve the script to create the database. To do this, select **CREATE To**, and then specify the destination of the script.

The output of these steps results in XML content that contains all metadata objects within the database and that conforms to the XML Schema definition language (XSD) documented in [\[MS-SSAS\]](#).

2.1.3.2 Using Analysis Management Objects

The Analysis Management Objects (AMO) object model enables implementers to programmatically manage a running instance of an Analysis Services database. For more information about AMO, see [\[MSDN-AMO\]](#).

To extract metadata by using AMO, follow these steps:

1. Use the **Server.Connect()** method to connect to Analysis Services.
2. Initialize a **System.Xml.XmlWriter** instance, such as **System.Xml.XmlTextWriter**.
3. Use the **Server.Connect()** method to connect to Analysis Services.
4. Flush and close the **System.Xml.XmlWriter** instance.

The output of these steps results in XML content that contains all metadata objects within the database and that conforms to the XSD language documented in [\[MS-SSAS\]](#).

2.1.3.3 Preconditions

To extract the metadata from an Analysis Services database, an Analysis Services server must have the database loaded, and this database must be accessible to the security principal executing the extraction commands.

2.1.3.4 Versioning

None.

2.1.3.5 Error Handling

None.

2.1.3.6 Coherency Requirements

There are no special coherency requirements.

2.1.3.7 Additional Considerations

There are no additional considerations.

2.1.3.7.1 Data Source Connection String

Because of security considerations, any explicit password that is specified in data source connection strings that are sent to Analysis Services cannot be retrieved and must be respecified.

2.2 Exporting Writeback Data

Certain Analysis Services deployments enable interactive updates to dimensions and **partition** data. Writeback features are supported only in Corporate BI mode and are not supported in Self-Service BI mode.

2.2.1 Data Description

2.2.1.1 Dimension Writeback

Dimension writebacks allow implementers to change, move, add, and delete attribute members within a dimension. These updates are stored directly in the data source table, which serves as the source for the dimension. For exporting this data, applications can query the underlying data source directly.

For more information about dimension writeback, see [\[MSFT-WBDIM\]](#).

2.2.1.2 Cube and Partition Writeback

Cube writebacks enable implementers to change **measure** data that is stored in a partition.

Once an implementer enables a cube for writeback through Business Intelligence Development Studio, Analysis Services performs the following operations:

- Creates a writeback table in the underlying data source that stores changes made by the end user as a difference from the current value. For example, if an end user changes a cell value from 90 to 100, the value +10 is stored in the writeback table, along with the time of the change and information about the end user who made the change.
- Creates a writeback partition within the cube that corresponds to the writeback table.

The net effect of accumulated changes is displayed to client applications. The original value in the cube is preserved, and an audit trail of changes is recorded in the writeback table.

For more information about write-enabled partitions, see [\[MSFT-WBPT\]](#).

2.2.2 Format and Protocol Summary

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

Protocol or format name	Description	Reference
SQL Server Analysis Services Protocol	Specifies methods for a client to communicate with, and perform operations on, an analysis server.	[MS-SSAS]

2.2.3 Data Portability Methodology

Writeback partitions can be identified through SQL Server Analysis Services Protocol [\[MS-SSAS\]](#) or through AMO [\[MSDN-AMO\]](#).

- In SQL Server Analysis Services Protocol, writeback partitions are **Partition** elements that have their **Type** element set to **Writeback**.
- In AMO, writeback partitions are objects of type **Partition** that have their **Type** property set to **PartitionType.Writeback**.

Because the data source bindings of a writeback partition are similar to those of a regular partition, information about the table that is used for storing writeback data can be retrieved through the **Source** property of the writeback partition. In the **TableBinding** type, the table name is stored in the **DbTableName** property.

2.2.3.1 Preconditions

To retrieve information about writeback partitions, the partitions must be accessible on an Analysis Services server by the security principal issuing discovery commands.

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.

3 Change Tracking

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

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
1.1 Glossary	Moved the term "cube" to [MS-GLOS] and added a citation to [MS-OFCGLOS].	N	Content updated.
1.2 References	Added reference [MS-OFCGLOS].	N	Content updated.

4 Index

C

[Change tracking](#) 11

G

[Glossary](#) 4

R

[References](#) 4

T

[Tracking changes](#) 11