SAP BusinessObjects Business Intelligence platform
Document Version: 4.2 – 2015-11-12

Using SAP NetWeaver BW in the Universe Design Tool

Content

1 Document History. 5

2 Using SAP NetWeaver BW in the universe design tool. 6

3 Prerequisites to using SAP BW in the universe design tool. 7

4 SAP Business Warehouse (BW) data sources. 8

4.1 SAP Business Warehouse (BW) InfoCubes as data sources. 8

4.2 SAP BW Queries as data sources. 9

4.3 Queries as recommended data sources. 10

4.4 SAP BW multilingual universes. 11

5 About OLAP universes. 12

5.1 What is an OLAP universe?. 12

5.2 Which OLAP data sources can be used to create a universe?. 13

SAP Business Warehouse (BW) data sources. 13

MSAS features supported for OLAP universes. 17

Essbase features supported for OLAP universes. 18

6 Defining connections to OLAP data sources. 20

6.1 About connections to OLAP data sources. 20

6.2 To start the New Connection wizard. 21

6.3 To select database middleware for an OLAP connection. 21

6.4 Login parameters for SAP BW OLAP connections. 22

6.5 Login parameters for MSAS OLAP connections 23

6.6 Defining login parameters for Essbase connections. 24

6.7 To select source cube or query for OLAP connections. 25

6.8 To define configuration parameters for OLAP connections. 26

6.9 Defining custom parameters for Essbase connections. 26

7 Customizing OLAP universes. 28

7.1 Creating OLAP Universes with additional parameters. 28

7.2 Defining OLAP options for your OLAP universe. 29

7.3 Defining objects in OLAP universes. 29

7.4 Universe design tool features supported for OLAP universes. 30

7.5 Database delegated projection function. 31

7.6 Setting delegated measures for OLAP universes. 32

7.7 Setting aggregate projection for a measure. 33

7.8 Calculated measures in OLAP universes. 34

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

To create a calculated measure in an OLAP universe. . . . 35
7.9 About MDX functions for cube queries. . . . 35
7.10 XML syntax for filter and WHERE statements. . . . 36
7.11 Predefined conditions in OLAP universes. . . . 37
XML Syntax for predefined filter options. . . . 37
Manually creating pre-defined conditions in an OLAP universe. . . . 38
About the Pre-defined Filter editor. . . . 39
About the options for the Pre-defined Filter editor. . . . 40
Editing a pre-defined filter with the pre-defined filter editor. . . . 41
7.12 Optional prompts in OLAP universes. . . . 42
7.13 To improve performance of certain queries on SAP BW universes. . . . 42

8 OLAP universe lifecycle management. . . . 44
8.1 About OLAP universe lifecycle management. . . . 44
8.2 Overview: relationship between universe object status and OLAP object status . . . 45
8.3 To refresh an OLAP universe. . . . 47
8.4 Regenerating Level 00 for OLAP universes. . . . 49
8.5 Renaming level L00 to ALL. . . . 49
8.6 Replacing OLAP universe level prefixes. . . . 50
8.7 Synchronizing the universe and the OLAP cube. . . . 50
8.8 How dimensions are managed in OLAP universe updates. . . . 50
When a dimension is unchanged. . . . 51
When a dimension is updated (name, description) . . . 52
When a dimension is deleted . . . 52
When a dimension is moved . . . 53
When a hierachy or characteristic is created . . . 54
When a dimension is new . . . 54
8.9 How hierarchies or characteristics are managed in OLAP universe updates . . . 55
When a hierarchy or characteristic is unchanged. . . . 55
When a characteristic business name or description is updated . . . 56
When a characteristic's active hierarchy is changed . . . 57
When a characteristic display attribute is changed to a navigation attribute. . . . 58
When a characteristic navigation attribute is changed to a display attribute . . . 59
When a hierarchy or characteristic is deleted . . . 60
When a hierarchy or characteristic is moved . . . 60
When a hierarchy or characteristic is new. . . . 61
8.10 How levels are managed in OLAP universe updates . . . 62
When a level is unchanged . . . 62
When the name or description of a level is updated . . . 63
When a level is deleted . . . 63
When a level is moved . . . 64
When a level is new . . . 65

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

8.11 How SAP variables are managed in OLAP universe updates . . . 65
When an SAP variable is unchanged. . . . 65
When the name or description of an SAP variable is updated . . . 66
When an SAP variable is deleted . . . 67
When an SAP variable is new . . . 68
8.12 How key figures or measures are managed in OLAP universe updates . . . 68
When a key figure or measure is unchanged . . . 69
When the name, description, or data type of a key figure or measure is updated . . . 69
When a key figure or measure is deleted . . . 70
When a key figure or measure is moved . . . 71
When a key figure or measure is new . . . 72
8.13 How SAP key dates are managed in OLAP universe updates . . . 72
When a SAP key date is unchanged . . . 72
When a SAP key date is deleted. . . . 73
When a SAP key date is new. . . . 73

9 How the different OLAP cubes are mapped to universes. . . . 75
9.1 How SAP BW objects are mapped and used in a universe. . . . 75
How characteristics are mapped and used. . . . 76
How key figures are mapped and used. . . . 77
How hierarchies are mapped and used. . . . 77
How variables are supported in universes. . . . 78
9.2 How Essbase cubes are mapped to universe components. . . . 83
9.3 How MSAS cubes are mapped to universe components . . . 84

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

1 Document History

Version	Date	Description
SAP BusinessObjects Business Intelligence platform 4.2	November 2015	Updated the guide with branding changes.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

2 Using SAP NetWeaver BW in the universe design tool

This guide is intended to help you use the universe design tool to create and manage OLAP universes based on SAP NetWeaver BW data sources. The universes can be used in Web Intelligence, Dashboard and Analytics, Live Office, and Query as a Web Service.

> **Note**
>
> The universe design tool creates .unv universes. An OLAP .unv universe cannot be converted to a .unx universe (.unx universes are created and modified using the information design tool).

An OLAP universe is a Business Objects universe that has been generated from an OLAP cube or query. The universe is created automatically from a selected connection to an OLAP data source.

You generate and maintain an OLAP universe in the following way:

- To generate an OLAP universe, first select the SAP NetWeaver BW data source.
- Define a connection to the data source using the New Connection wizard, and select the connection for a new universe.
  Designer generates the universe automatically. OLAP structures are mapped directly to classes, measures, dimensions, details, and filters in the universe. The universe structure appears in the Universe pane.
- You can save and export the OLAP universe to the CMS.
- You can modify any of the OLAP universe components.
- The Update OLAP Universe wizard allows you manage the lifecycle of the OLAP universe. The wizard automatically refreshes the universe structure with changes made in the OLAP data source. The wizard can differentiate generated objects from objects added or modified manually, allowing you to preserve the manual changes made in Designer.

> **Note**
>
> This guide sometimes refers to other OLAP data sources, or OLAP data sources in general. In these cases, the information applies equally to SAP NetWeaver BW data sources.

Related Information

Prerequisites to using SAP BW in the universe design tool [page 7]
SAP Business Warehouse (BW) data sources [page 8]
Universe design tool features supported for OLAP universes [page 30]
About OLAP universe lifecycle management [page 44]
How SAP BW objects are mapped and used in a universe [page 75]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

3 Prerequisites to using SAP BW in the universe design tool

When creating a universe from an SAP BW data source, you can enable SSO (Single Sign On) for view time. SSO allows users to log on to SAP BusinessObjects Enterprise with their SAP Credentials and leverage SAP Authentication.

To enable SSO for OLAP Universes on top of SAP, you must install the SAP Integration and configure the SAP Security Plug In.

Once SAP security integration is configured, you can use SAP credentials to launch universe design tool. Create the BusinessObjects Enterprise user name by concatenating the SAP System ID and SAP Client ID as defined when configuring the security integration with the SAP user ID.

For more information, see the *Business Objects XI Integration for SAP Solutions Installation Guide*, and the *Business Objects XI Integration for SAP Solutions User Guide*.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

4 SAP Business Warehouse (BW) data sources

When creating an OLAP universe based on a BW data source, you can build the universe based directly on an InfoCube/MultiCube, or based on a BEx Query enabled on top of any InfoProvider. An InfoProvider can be:

an InfoCube
a MultiCube or Multi-InfoProvider
an Operational Data Store (ODS)
an InfoSet

Related Information

SAP Business Warehouse (BW) InfoCubes as data sources [page 8]
SAP BW Queries as data sources [page 9]
Queries as recommended data sources [page 10]

4.1 SAP Business Warehouse (BW) InfoCubes as data sources

The following types of InfoCubes are supported as data sources for building OLAP universes:

- Standard and Transactional InfoCubes: Data and metadata are physically stored in the same SAP Business Warehouse (BW) system
- Remote InfoCube: Data is physically stored on a remote system

> **Note**
>
> While fully supported, building and deploying universes on remote InfoCubes is not recommended for ad-hoc query-, reporting-, and analysis-use scenarios. Such architecture is generally not expected to meet query performance expectations with interactive queries.

- MultiCubes and Multi-InfoProviders

> **Note**
>
> Building and deploying a Business Objects universe on top of a MultiCube or Multi-InfoProvider is identical to building and deploying a universe on top of an InfoCube.

All the characteristics, hierarchies, key figures, including time and unit, in the InfoCube are visible in the universe.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

4.2 SAP BW Queries as data sources

SAP BW customers use BEx Queries to access SAP Business Explorer front-ends.

> **Note**
>
> In order to serve as a data source and become available through the OLAP interface to Business Objects universes, BEx queries must be released for OLE DB for OLAP. You allow external access to the BEx Query in the SAP BW Query Designer, on the *Extended mode* tab of the *Query Properties* dialog box.

All InfoObjects in the BEx query selected as rows, columns, and free characteristics are visible in the universe. This includes characteristics, hierarchies, key figures, structures, and variables.

Both InfoSets and Operational Data Stores (ODS) can be exposed to universes via BEx Queries.

Queries based on an ODS

An ODS can be exposed to a universe via a BEx Query.

ODS objects are often used to manage detailed transaction-level data before it is aggregated into InfoCubes. Including ODS objects in the SAP NetWeaver technology platform data store design is a way to minimize InfoCube size and improve loading and querying performance.

> **Note**
>
> An ODS is usually a large, detailed relational structure. Accessing an ODS via the OLAP BAPI interface does not deliver ideal query performance. Consider these alternatives to meet end-user expectations for fast report delivery:

- Create direct access to an ODS via BAPI calls
- Access the ODS tables using direct SQL in Web Intelligence

Queries based on an InfoSet

An InfoSet can be exposed to a universe via a BEx Query.

InfoSets are sometimes defined in SAP BW to report master data.

> **Note**
>
> You can report master data by basing the universes on InfoCubes, eliminating the requirement to go through InfoSets and BEx Queries. The key difference between the two approaches is that master data reported off InfoCubes limits data to valid transactions.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Related Information

Queries as recommended data sources [page 10]

4.3 Queries as recommended data sources

BEx Queries are recommended as data sources for generating Business Objects universes for the following reasons:

- Not all SAP BW metadata features can be retrieved on an InfoCube level, as summarized in the following table.

  Table 1:

BW metadata feature	SAP OLAP Business Application Programming Interface (BAPI) support level
Characteristics (incl. Time and Unit)	InfoCube/BEx Query
Hierarchies	InfoCube/BEx Query
Basic Key Figures	InfoCube/BEx Query
Navigational Attributes	BEx Query only
Display Attributes	InfoCube/BEx Query
Calculated Key Figures / Formulas	BEx Query only
Restricted Key Figures	BEx Query only
Custom Structures	BEx Query only
Variables	BEx Query only

- BEx Queries offer a flexible extension to the data modeling environment. InfoCubes require more effort to change.
- BEx Queries offer significant functionality to create customized data sources that meet end-user requirements.

Although BEx Queries have advantages as data sources, you do not need a BEx Query for every report, nor do you need a universe for every existing BW Query. To minimize maintenance costs, focus the implementation strategy on limiting the final number of BEx Queries and universes required to meet all the ad-hoc query and reporting needs. Keep in mind the following points to reduce the number of universes needed:

When Web Intelligence is the front-end tool, you are not restricted by the output format in the BEx Query.

There is no direct impact on performance when working with OLAP universes created from large BEx Queries. OLAP universe objects not inserted in the Web Intelligence query have no direct impact on the query performance.

Note

Business Objects recommends having a few BEx Queries – from a single one to a handful of them – for every InfoCube or MultiCube that is in scope for ad-hoc query and reporting. Then build a universe on top of each of these BEx Queries.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

4.4 SAP BW multilingual universes

With Web Intelligence, it is possible to leverage the multilingual capabilities of SAP BW. In order to implement a multilingual environment, the BW system must include multilingual metadata and multilingual data.

You must create a universe for each language supported by the solution. The language in which the universe connection is created determines the language in which the universe is generated.

The user's SAP authentication determines the language of the data returned to the query. The user must log into InfoView using SAP authentication and specify the desired language for results returned from the SAP server.

The result-set language is dependent on SAP's Unicode support. If the SAP system does not contain the data in the desired language, the data is not available in Web Intelligence in this language. Web Intelligence reverts to displaying technical names instead of descriptions when the descriptions are not translated in BW.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

5 About OLAP universes

5.1 What is an OLAP universe?

An OLAP universe is a Business Objects universe that has been generated from an OLAP cube or query. The universe is created automatically from a selected connection to an OLAP data source.

Once the universe has been created, it can be exported to the Central Management Server (CMS) as any other universe. The universe is then available to Web Intelligence users to run queries and create reports.

You generate and maintain an OLAP universe in the following way:

- To generate an OLAP universe, first select the OLAP data source.

  > **Note**
  >
  > For a secured connection to an OLAP data source, any users who need to generate the universe or refresh its structure need to have the *Download* right on the connection. This right is set in the CMC by the administrator.

- Define a connection to the data source using the New Connection wizard, and select the connection for a new universe.
  The universe design tool generates the universe automatically. OLAP structures are mapped directly to classes, measures, dimensions, details, and filters in the universe. The universe structure appears in the Universe pane.
- You can save and export the OLAP universe to the CMS.
- You can modify any of the OLAP universe components.
- The Update OLAP Universe wizard allows you manage the lifecycle of the OLAP universe. The wizard automatically refreshes the universe structure with changes made in the OLAP data source. The wizard can differentiate generated objects from objects added or modified manually, allowing you to preserve the manual changes made in the universe design tool.

Related Information

Which OLAP data sources can be used to create a universe? [page 13]
About connections to OLAP data sources [page 20]
Universe design tool features supported for OLAP universes [page 30]
About OLAP universe lifecycle management [page 44]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

5.2 Which OLAP data sources can be used to create a universe?

You can create OLAP universes automatically from the following OLAP data sources:

- SAP Business Warehouse (BW)
- Microsoft Analysis Services (MSAS) 2000
- Microsoft Analysis Services (MSAS) 2005
- Hyperion Essbase

> **Note**
>
> In order to connect to an Essbase OLAP data source from SAP BusinessObjects OLAP products including universe design tool, Web Intelligence Rich Client and Web Intelligence, ensure that Essbase Client middleware is properly installed and configured on machines hosting those SAP BusinessObjects OLAP products. Specifically, ensure that Essbase Client environment variables ARBORPATH and ESSBASEPATH are created and set as Windows system environment variables (as opposed to Windows user environment variables).

One universe is generated automatically from one cube or query. OLAP universes support a single cube in the universe.

Related Information

SAP Business Warehouse (BW) data sources [page 8]
How SAP BW objects are mapped and used in a universe [page 75]
MSAS features supported for OLAP universes [page 17]
How MSAS cubes are mapped to universe components [page 84]
Essbase features supported for OLAP universes [page 18]
How Essbase cubes are mapped to universe components [page 83]

5.2.1 SAP Business Warehouse (BW) data sources

When creating an OLAP universe based on a BW data source, you can build the universe based directly on an InfoCube/MultiCube, or based on a BEx Query enabled on top of any InfoProvider. An InfoProvider can be:

- an InfoCube
- a MultiCube or Multi-InfoProvider
- an Operational Data Store (ODS)
- an InfoSet

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Related Information

SAP Business Warehouse (BW) InfoCubes as data sources [page 8]
SAP BW Queries as data sources [page 9]
Queries as recommended data sources [page 10]

5.2.1.1 SAP Business Warehouse (BW) InfoCubes as data sources

The following types of InfoCubes are supported as data sources for building OLAP universes:

- Standard and Transactional InfoCubes: Data and metadata are physically stored in the same SAP Business Warehouse (BW) system
- Remote InfoCube: Data is physically stored on a remote system

> **Note**
>
> While fully supported, building and deploying universes on remote InfoCubes is not recommended for ad-hoc query-, reporting-, and analysis-use scenarios. Such architecture is generally not expected to meet query performance expectations with interactive queries.

- MultiCubes and Multi-InfoProviders

> **Note**
>
> Building and deploying a Business Objects universe on top of a MultiCube or Multi-InfoProvider is identical to building and deploying a universe on top of an InfoCube.

All the characteristics, hierarchies, key figures, including time and unit, in the InfoCube are visible in the universe.

5.2.1.2 SAP BW Queries as data sources

SAP BW customers use BEx Queries to access SAP Business Explorer front-ends.

> **Note**
>
> In order to serve as a data source and become available through the OLAP interface to Business Objects universes, BEx queries must be released for OLE DB for OLAP. You allow external access to the BEx Query in the SAP BW Query Designer, on the *Extended mode* tab of the *Query Properties* dialog box.

All InfoObjects in the BEx query selected as rows, columns, and free characteristics are visible in the universe. This includes characteristics, hierarchies, key figures, structures, and variables.

Both InfoSets and Operational Data Stores (ODS) can be exposed to universes via BEx Queries.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Queries based on an ODS

An ODS can be exposed to a universe via a BEx Query.

ODS objects are often used to manage detailed transaction-level data before it is aggregated into InfoCubes. Including ODS objects in the SAP NetWeaver technology platform data store design is a way to minimize InfoCube size and improve loading and querying performance.

> **Note**
>
> An ODS is usually a large, detailed relational structure. Accessing an ODS via the OLAP BAPI interface does not deliver ideal query performance. Consider these alternatives to meet end-user expectations for fast report delivery:

- Create direct access to an ODS via BAPI calls
- Access the ODS tables using direct SQL in Web Intelligence

Queries based on an InfoSet

An InfoSet can be exposed to a universe via a BEx Query.

InfoSets are sometimes defined in SAP BW to report master data.

> **Note**
>
> You can report master data by basing the universes on InfoCubes, eliminating the requirement to go through InfoSets and BEx Queries. The key difference between the two approaches is that master data reported off InfoCubes limits data to valid transactions.

Related Information

Queries as recommended data sources [page 10]

5.2.1.3 Queries as recommended data sources

BEx Queries are recommended as data sources for generating Business Objects universes for the following reasons:

- Not all SAP BW metadata features can be retrieved on an InfoCube level, as summarized in the following table.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Table 2:

BW metadata feature	SAP OLAP Business Application Programming Interface (BAPI) support level
Characteristics (incl. Time and Unit)	InfoCube/BEx Query
Hierarchies	InfoCube/BEx Query
Basic Key Figures	InfoCube/BEx Query
Navigational Attributes	BEx Query only
Display Attributes	InfoCube/BEx Query
Calculated Key Figures / Formulas	BEx Query only
Restricted Key Figures	BEx Query only
Custom Structures	BEx Query only
Variables	BEx Query only

- BEx Queries offer a flexible extension to the data modeling environment. InfoCubes require more effort to change.
- BEx Queries offer significant functionality to create customized data sources that meet end-user requirements.

Although BEx Queries have advantages as data sources, you do not need a BEx Query for every report, nor do you need a universe for every existing BW Query. To minimize maintenance costs, focus the implementation strategy on limiting the final number of BEx Queries and universes required to meet all the ad-hoc query and reporting needs. Keep in mind the following points to reduce the number of universes needed:

When Web Intelligence is the front-end tool, you are not restricted by the output format in the BEx Query.

There is no direct impact on performance when working with OLAP universes created from large BEx Queries. OLAP universe objects not inserted in the Web Intelligence query have no direct impact on the query performance.

Note

Business Objects recommends having a few BEx Queries – from a single one to a handful of them – for every InfoCube or MultiCube that is in scope for ad-hoc query and reporting. Then build a universe on top of each of these BEx Queries.

5.2.1.4 SAP BW multilingual universes

With Web Intelligence, it is possible to leverage the multilingual capabilities of SAP BW. In order to implement a multilingual environment, the BW system must include multilingual metadata and multilingual data.

You must create a universe for each language supported by the solution. The language in which the universe connection is created determines the language in which the universe is generated.

The user's SAP authentication determines the language of the data returned to the query. The user must log into InfoView using SAP authentication and specify the desired language for results returned from the SAP server.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

The result-set language is dependent on SAP's Unicode support. If the SAP system does not contain the data in the desired language, the data is not available in Web Intelligence in this language. Web Intelligence reverts to displaying technical names instead of descriptions when the descriptions are not translated in BW.

5.2.1.5 Prerequisites to using SAP BW in the universe design tool

When creating a universe from an SAP BW data source, you can enable SSO (Single Sign On) for view time. SSO allows users to log on to SAP BusinessObjects Enterprise with their SAP Credentials and leverage SAP Authentication.

To enable SSO for OLAP Universes on top of SAP, you must install the SAP Integration and configure the SAP Security Plug In.

Once SAP security integration is configured, you can use SAP credentials to launch universe design tool. Create the BusinessObjects Enterprise user name by concatenating the SAP System ID and SAP Client ID as defined when configuring the security integration with the SAP user ID.

For more information, see the *Business Objects XI Integration for SAP Solutions Installation Guide*, and the *Business Objects XI Integration for SAP Solutions User Guide*.

5.2.2 MSAS features supported for OLAP universes

The following table summarises the support level for MSAS features for universes generated from an MSAS data source.

MSAS metadata feature	OLAP universe support level
Cube	Supported
Local cube	Supported
Virtual cube (MSAS 2000)	Supported
Perspective (MSAS 2005)	Supported
Dimensions	Supported
Virtual dimensions (MSAS 2000)	Supported
Hierarchies	Supported
Levels	Supported
Level Property	Supported
Attributes (MSAS 2005)	Supported
Measures	Supported
Measure group (MSAS 2005)	Supported
Calculated measures	Supported

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

MSAS metadata feature	OLAP universe support level
Display folder (MSAS 2005)	Supported
KPI (MSAS 2005)	Not supported
Action	Not supported
Database sort order	Must define custom sort order in Web Intelligence
Write back	Not supported

Related Information

How MSAS cubes are mapped to universe components [page 84]

5.2.3 Essbase features supported for OLAP universes

The following table summarizes the support level for Essbase features for universes generated from a Hyperion Essbase data source.

Essbase metadata feature	OLAP universe support level
Block storage mode	Supported
Aggregate storage mode	Supported
Hybrid mode	Not supported
Alias tables	Supported
Dimensions	Supported
Attribute dimensions	Supported
Duplicate members	Supported
Generations	Supported
Levels	Not supported
User Define Attributes (UDA)	Not supported
Dynamic Time Series (DTS)	Not supported
Essbase Integration Services (EIS) Drill-through	Not supported
Substitution variables	Not supported
Linked partitions	Not supported
Linked Reporting Objects (LRO)	Not supported
Database sort order	Must define custom sort order in Web Intelligence
Write back	Not supported

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Related Information

How Essbase cubes are mapped to universe components [page 83]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

6 Defining connections to OLAP data sources

6.1 About connections to OLAP data sources

To generate an OLAP universe, you must first define a connection to the OLAP data source. Define one connection for each cube or query that you want to use to create a universe.

Use the New Connection wizard to define the connection. The wizard leads you through the following steps in creating a connection:

- Starting the New Connection wizard in the universe design tool.
- Naming the connection and selecting the database middleware.
- Defining the login parameters for the connection. These parameters vary depending on the database middleware that you selected.
- Selecting the cube or query to use to create the universe.
- Defining the connection life-time.
- Defining custom parameters. These parameters vary depending on the database middleware that you selected.

Defining a connection is the first step in creating an OLAP universe. Once you have defined the connection, the universe design tool generates the universe automatically.

> **Note**
>
> If you define a connection from the connections list on the Tools menu, you must create the universe as a separate step.

Related Information

To start the New Connection wizard [page 21]
To select database middleware for an OLAP connection [page 21]
Login parameters for SAP BW OLAP connections [page 22]
Login parameters for MSAS OLAP connections [page 23]
Defining login parameters for Essbase connections [page 24]
To select source cube or query for OLAP connections [page 25]
To define configuration parameters for OLAP connections [page 26]
Defining custom parameters for Essbase connections [page 26]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

6.2 To start the New Connection wizard

To start the New Connection wizard, do one of the following:

Start from...	Do...
New Universe icon	Click the *New Universe* icon, then click *New...* on the *Definition* page of the *Universe Parameters* box.
File menu	From an empty session, select *File* > *Parameters*, then click *New...* on the *Definition* page of the *Universe Parameters* box.
Quick Design wizard	If the Quick Design wizard is enabled, it starts automatically when you start the universe design tool. Click *New...* in step one of the Quick Design wizard. **Note** If the wizard has been disabled, select *File* > *New*. If the wizard does not start when you select *File* > *New*, select *Tools* > *Options*. On the *General* page of the *Options* dialog box, select the *File/New starts Quick Design wizard* check box. Click *OK* and select *File* > *New*.
Tools menu	Select *Tools* > *Connections*. Click *Add...* in the *Wizard Connection* dialog box.

6.3 To select database middleware for an OLAP connection

On the *Database Middleware Selection* page of the New Connection wizard, enter a name for the connection, select the type and the database middleware for the connection.

Database Middleware Selection parameter	Description
Connection Type	Select *Secured* for controlled access to the connection (recommended). Select *Shared* to allow uncontrolled access to the connection to all users. Select *Personal* to restrict access to the universe creator. You can use personal connections to access personal data on a local machine only.
Connection Name	Enter a name for the connection.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Database Middleware Selection parameter	Description
Filter Stored Procedures Network Layers	The *Filter Stored Procedures Network Layers* parameter is not used for OLAP connections.
List of available data access drivers.	The page lists the databases and middleware that correspond to your data access driver key. Expand the node for your target database to show the supported middleware for that database. Expand the middleware node to show the Business Objects data access driver for the OLAP middleware. Select the data access driver.

6.4 Login parameters for SAP BW OLAP connections

The *New Connection* wizard's *Login Parameters* dialog box can contain the following parameters:

Table 3:

Login parameter	Description
Authentication mode	• *Use specified user name and password*: uses the login details as authentication. • *Use Business Objects credential mapping*: the user is prompted for a database user password associated with his BusinessObjects account to refresh a report. This is set using the parameters *dbuser* and *dbpass*. These are set at the administrative level. Refer to *SAP Business Objects Enterprise Administrator's Guide* for information on setting up this option. • *Use Single Sign On when refreshing reports at View Time*: When selected, the username and password used to access the CMS are automatically used as the database login parameters. See the *Business Objects Enterprise Administration Guide* for information on setting Single Sign-On (SSO).
Use SNC if available	Select this check box to use the SNC.
Client	The number used to identify the client on SAP BW system (required).
User name	The user name to use to access the OLAP server if the *Authentication mode* is *Use specified user name and password*.
Password	The password to use to access the OLAP server if the *Authentication mode* is *Use specified user name and password*.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Login parameter	Description
Language	The language that will be used for the connection. **Note** The connection language determines the language in which the universe is generated.
Save Language	Which language will be used for the connection: • If you check *Save Language*, the value from the *Language* field will be used. • If you clear *Save Language*, the value from the user's session will be used.
Login mode or *Server Type*	Select *Application Server* to connect directly to the SAP server without using load balancing. Select *Message Server* to benefit from SAP load balancing capabilities.
Application Server	Select or enter the name or IP address of the SAP application server (required for Application server login mode).
System Number	Enter the system number, for example 00 (required for Application server login mode) .
System ID	Enter the *Message Server*, *Logon Group*, and if required, the *System ID* when using *Message Server* login mode.
Logon Group	
Message Server	

6.5 Login parameters for MSAS OLAP connections

The *New Connection* wizard's *Login Parameters* dialog box can contain the following parameters:

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Table 4:

Login parameter	Description
Authentication mode	• *Use specified user name and password*: uses the login details as authentication. • *Use Business Objects credential mapping*: the user is prompted for a database user password associated with his BusinessObjects account to refresh a report. This is set using the parameters *dbuser* and *dbpass*. These are set at the administrative level. Refer to *SAP Business Objects Business Intelligence platform Administrator's Guide* for information on setting up this option. • *Use Single Sign On when refreshing reports at View Time*: When selected, the username and password used to access the CMS are automatically used as the database login parameters. See the *SAP Business Objects Business Intelligence platform Administrator's Guide* for information on setting Single Sign-On (SSO).
Server	Enter one of the following: • The URL for the MSAS library exposed and configured in the MSAS server. • The server name for the MSAS data source • The full path file name of an MSAS cube file. Enter the full path file name between double quotes, for example: "Z:\All cubes\test.cub" **Note** If the cube file is located on a host system different from the SAP BusinessObjects host, the two machines must have a shared connection. You must create the connection to a cube file directly on the SAP BusinessObjects host.
User name	The user name to use to access the OLAP server if the *Authentication mode* is *Use specified user name and password*.
Password	The password to use to access the OLAP server if the *Authentication mode* is *Use specified user name and password*.
Language	The language that will be used for the connection.

6.6 Defining login parameters for Essbase connections

On the *Login Parameters* page of the New Connection wizard, specify the login details for connecting to the Essbase database.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Table 5:

Login parameter	Description
Authentication mode	Select *Use specified username and password* to require the user to enter login information when using the connection. To synchronize Essbase security with BusinessObjects Enterprise, enter the Essbase DBuser and DBpass for *User name* and *Password*. Select *Use BusinessObjects credential mapping* to use the user's BusinessObjects Enterprise login credentials for the connection. **Note** The *Use Single Sign On when refreshing reports at view time* option is not supported for Essbase connections.
User name	Enter the Essbase DBuser.
Password	Enter the Essbase DBpass.
Server	Enter the Essbase server name.

6.7 To select source cube or query for OLAP connections

The cube browser shows the OLAP cubes available to the target server.

Expand the cube nodes to display the cubes and queries available. The browser has the following tools to help you search:

Table 6:

Cube browser tool	Description
Favorites	A folder that holds links to cubes that you select for quick access. To add a cube to Favorites, right click a cube in the OLAP cube browser and select Add to Favorites from the contextual menu.
Search	Searches the names of available cubes or queries for a text string. Type a string in the text box and click Search. Each found instance is highlighted. Click Search to continue searching.
$INFOCUBE folder	For SAP BW data sources, the InfoCubes and MultiCubes are grouped in a folder called $INFOCUBE.

Select the cube or query to use to create the universe.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

6.8 To define configuration parameters for OLAP connections

On the *Configuration Parameters* page of the New Connection wizard, define the connection parameters to manage the connection life-time. You can accept the default settings when you create the connection and modify them later.

Table 7:

Configuration parameter	Description
Connection Pool Mode	You define the life-time with the *Connection Pool Mode* and *Pool timeout* parameters. By default, the *Connection Pool Mode* is set to *Keep the connection alive for*, with a default *Pool timeout* of 10 minutes. **Note** Business Objects recommends you keep the default connection life-time. The universe building process slows significantly if the *Connection Pool Mode* is set to disconnect after each transaction. Disconnecting after each transaction also impacts key end-user workflows such as working with hierarchical list of values. The connection life-time can have a significant impact when working with SAP BW. However, connection life-time can also impact updating existing universes with changes in the BEx Query. This is because the OLAP BAPI interface builds a metadata cache on the client side every time a connection to SAP BW is established. This cache is only emptied when the connection closes. To minimize the risk of metadata cache being desynchronized with SAP BEx Query updates, you can change the *Pool timeout* from 10 minutes to 1 minute. When working in parallel editing BW Queries and mapping new universes to these queries, it is recommended that you close the universe design tool (so that universe connections are also closed and the metadata cache is emptied) before building any new universes to take into account changes that were just made on the BEx Query side.
Pool timeout	
Array fetch size	The *Array fetch size* parameter allows you to set the optimum number of rows that are permitted in a FETCH procedure.
Array bind size	The *Array bind size*, and *Login timeout* parameters are not used for OLAP connections.
Login timeout	

6.9 Defining custom parameters for Essbase connections

On the *Custom Parameters* page of the New Connection wizard, specify the alias table and select the dimension to be used as the Measure dimension when generating the universe.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Table 8:

Login parameter	Description
Alias table	To generate the universe on an alias table other than the default, select the alias table from the list.
Measure dimension	Select the dimension to use as the Measure dimension. the universe design tool generates the members of the dimension you select as measures in the universe.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

7 Customizing OLAP universes

7.1 Creating OLAP Universes with additional parameters

This feature applies to OLAP universes only and allows you to define additional metadata parameters when you create an OLAP universe with MSAS, SAP Business Warehouse (BW), or Essbase.

When you create an OLAP universe you can define the following parameters:

Generic OLAP options	Description
Generate technical names as details	You can set the application to generate the technical name as a detail object of the dimensions in the universe. When the universe is generated, this creates detail objects that point to the technical names.

SAP OLAP options	Description
Set measures aggregation to delegated	You can set the application to set the aggregation function of measures to database delegated.
Replace prefixes L00, L01,	Universe level prefixes indicate the level in the hierarchy of an object. Level L00 is the top or root level, L01 is the next level down. In the "New Universe Wizard", you can replace OLAP universe level prefixes with a different prefix. The level numbering is retained, but the prefix 'L' can be replaced by Level, for example. Type your own prefix in the *New prefix* field. This prefix is prepended to all levels throughout the OLAP universe.
Rename level 00 to All	This option is disabled if *Generate level 00* is set to *No*. You can rename the top level (root level) L00 to All the next time the universe is generated.
Generate level 00	This option only applies to SAP Characteristics. You can disactivate this option for Characteristics and hierarchies. Level 00 is always generated for hierarchy variables. You can regenerate the Level numbers (L00, L01, L02...) when you generate or update a universe. The level numbers are appended to the level names (e.g. "Monthly Sales_L01"). This is useful for Web Intelligence reports, where the All level is used to aggregate results for a query. This avoids having to create the aggregate field in the Web Intelligence report.

Note

When the universe is created with "Generate Level 00" disabled, root level is not generated for hierarchies.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

7.2 Defining OLAP options for your OLAP universe

Use the OLAP options to define how specific universe metadata is generated from an OLAP source. You can select OLAP options from the *OLAP* page of the *Options* dialog box (*Tools* > *Options* > *OLAP*). All content of the OLAP source is extracted and created in the universe depending on selected options. You can select the following OLAP universe generation options:

Generic OLAP options	Description
Generate technical names as details	You can set the application to generate the technical name of the universe as a property. When the universe is generated, this creates an object that points to the technical name.

SAP OLAP options	Description
Set measures aggregation to delegated	You can set the application to generate delegated measures for measures that use the aggregation function. When the universe is generated, any measures using the aggregation function are set to database delegated.
Replace prefixes L00, L01,	Universe level prefixes indicate the level in the hierarchy of an object. Level L00 is the top or root level, L01 is the next level down. In the "New Universe Wizard", you can replace OLAP universe level prefixes with a different prefix. The level numbering is retained, but the prefix 'L' can be replaced by Level, for example. Type your own prefix in the *New prefix* field. This prefix is prepended to all levels throughout the OLAP universe.
Rename level 00 to All	This option is disabled if Generate level 00 is set to No. You can rename the top level (root level) L00 to ALL when the universe is next generated. This is useful for Web Intelligence reports, where the All level is used to aggregate results for a query. This avoids having to create the aggregate field in the Web Intelligence report.
Generate level 00	This option only applies to SAP Characteristics. You can disactivate this option for Characteristics only. Level 00 is always generated for hierarchies and hierarchy variables. You can regenerate the Level numbers (L00, L01, L02...) when you generate or update a universe. The level numbers are appended to the level names (e.g. "Monthly Sales_L01")

7.3 Defining objects in OLAP universes

You can use the SQL editor to define a Select statement or a Where clause for an object, and to insert MDX operators and functions for OLAP universe objects. The options and functions available in the SQL editor depend on the underlying database.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

7.4 Universe design tool features supported for OLAP universes

OLAP universes are created automatically. Once you have created the OLAP universe, you can modify any of the universe components.

These universe design tool features are supported for generated OLAP universes:

- Hide, duplicate, and rename classes and objects (dimension, detail, and measure)
- Insert new classes and objects (dimension, detail, and measure)
- Edit an object's format
- Edit an object's data type
- Define primary and foreign keys
- Parse dimension, detail, and measure object MDX syntaxes
- Check universe integrity
- Edit hierarchies
- Create cascading list of values
- Define a delegate search for a list of values, allowing users to limit loading of the list of values at query run time
- Use default values for variables
- Define measures with database delegated projection function (smart measures)
- Refresh the universe structure

In addition, the following features are available uniquely for OLAP universes:

- Create calculated measures (SAP BW and MSAS only)
- Create pre-defined conditions
- Define optional prompts

All objects based on OLAP universes are generated with index awareness. If there are duplicate values in a hierarchy for an object, index awareness eliminates inconsistency in the list of values. For example, say that Paris occurs twice in the hierarchy, once under the parent France and once under the parent Texas. If the user selects Paris under France, only rows for Paris, France are returned.

The following universe design tool features are not supported for OLAP universes:

- You cannot set row-level security authorizations in an OLAP universe.
- You cannot edit a list of values in an OLAP universe.
- You cannot view and edit the universe entity-relationship schema as no schema is generated for OLAP universes.

Related Information

Calculated measures in OLAP universes [page 34]
Predefined conditions in OLAP universes [page 37]
Optional prompts in OLAP universes [page 42]
Database delegated projection function [page 31]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

7.5 Database delegated projection function

In a universe, any measure can hold a projection function (*Sum*, *Min*, *Max*, *Count*, and *Avg*). The projection function is used to aggregate the measure locally in Web Intelligence when the number of dimensions displayed in a report is smaller than the number of dimensions in the query result set.

Non-additive measures, such as ratio, average, and weight, can only be shown at the same aggregation level as the query result set. Therefore, non-additive measures generally have their projection function set to *None* in the universe.

The projection function *Database delegated* allows you to delegate the aggregation of a non-additive measure to the database server. These are called smart measures in Web Intelligence. A smart measure has its projection function set to *Database delegated* on the properties page of the object properties. For more information about how these and other functions are used in Web Intelligence , please refer to section *Calculating values with Smart Measures* in the document *Using Functions, Formulas and Calculations in Web Intelligence*.

Note

For OLAP universes based on MSAS and Essbase data sources, all measures are created in the universe with the projection function set to *Database delegated* by default.

Note

Please be aware of the following limitation when using a smart measure based on a measure that has aggregate aware set: It is strongly recommended to ensure that the aggregate tables used in the measure definition have consistent data (the aggregate values are accurate regarding detail values) otherwise the smart measures can produce inconsistent data. For example, if a year aggregate table and a day aggregate table are used for a smart measure, the year aggregate table is consistent with the day aggregate table for complete years, but for the current year, the year table can be empty while the day table is accurate on a daily basis. In this case, a report that uses smart measures based on the current year and daily tables can give incoherent results.

Example

Smart measure

In this example, a query contains two dimensions: Country and Region, and three measures: Order Amount, Delivered Amount, and % of Delivered Amount.

L01 Region	Amount Delivered	Order Quantity	% Delivered
Reg1	497,318,880	497,332,680	99.997
Reg2	199,463,776	199,466,536	99.998
Reg3	198,927,552	198,933,072	99.997
		Sum:	299.992

The sum for % Delivered is incorrect because it is a summation of the % Delivered column.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

If this measure has a projection function set to *Database delegated* in the universe, when the user refreshes the report, Web Intelligence connects to the database to calculate the correct value.

L01 Region	Amount Delivered	Order Quantity	% Delivered
Reg1	497,318,880	497,332,680	99.997
Reg2	199,463,776	199,466,536	99.998
Reg3	198,927,552	198,933,072	99.997
		Sum:	299.992
		Total:	99.997

Note

The some functions such as the ratio function (Average) must be used with caution. When calculating an average from a column, the behavior of this function can be unexpected when it is not configured correctly.

For example, the SQL function `sum(Shop_facts.Margin)/sum(Shop.facts.Quantity_sold)` can have unexpected results. When incorrectly configured, it will calculate the average for each cell and return the sum of those averages. To correct this behavior, the parametrization of the function must be performed as follows:

1. Go to the *Edit Properties* option for the function.
2. For the option *Choose how this measure will be projected when aggregated*, select the Function *Db delegated* from the Function dropdown list.
3. Save your changes.

Related Information

Setting aggregate projection for a measure [page 33]

7.6 Setting delegated measures for OLAP universes

You can set the application to generate delegated measures for measures that use the aggregation function. When the universe is generated, any measures using the aggregation function are set to database delegated...

Related Information

Replacing OLAP universe level prefixes [page 50]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Regenerating Level 00 for OLAP universes [page 49]
Renaming level L00 to ALL [page 49]

7.7 Setting aggregate projection for a measure

When you create a measure you must specify the way the aggregate function will be projected onto a report.

Returned values for a measure object are aggregated at two levels of the query process:

- Query level. Data is aggregated using the inferred SELECT statement.
- Microcube to block level. When data is projected from the microcube to the block in a report. This projection function of measures allows local aggregation in the microcube.

> **Note**
>
> A microcube is a conceptual way to present the data returned by a query before it is projected onto a report. It represents the returned values held in memory by a Business Objects reporting product. The block level is the 2 dimensional report that a user creates with the returned data. A user can choose to use all, or only some of the data held in the microcube to create a report. A user can also do aggregate functions on the returned values in the microcube (local aggregation) to create new values on a report.

The two levels of aggregation fit into the query process as follows:

- User creates a query in Web Intelligence.
- Web Intelligence infers the SQL from the query and sends a SELECT statement to the target database.
- The data is returned to the microcube. This is the first aggregation level.
- The microcube projects the aggregated data onto the report. Data is split out in the Query pane requiring aggregation to lower levels. This is the second aggregation level.

When you initially make a query the result set of the Select statement is stored in the microcube, and all data then held in the microcube is projected into a block. As data is projected from the lowest level held in the microcube no projection aggregation is taking place.

However, when you use the Query pane to project only partial data from the microcube, aggregation is required to show measure values at a higher level.

For example, in the previous example, if you do not project the year data into the block, the three rows related to Year need to be reduced to one row to show the overall Sales Revenue for that resort, so a sum aggregation is used.

You set projection aggregation on the *Properties* page of the *Edit Properties* sheet for a measure (right-click Object > Object Properties > Properties).

Projection aggregation is different from SELECT aggregation.

Related Information

Database delegated projection function [page 31]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

7.8 Calculated measures in OLAP universes

You can create calculated measures in the universe to restrict queries. Calculated measures in OLAP universes are like definitions on the measure object in non-OLAP universes, except that you define the restriction using MDX functions embedded in XML tags rather than using SQL.

Calculated measures are available for these OLAP data sources:

- SAP Business Warehouse (BW)
- MSAS 2000 and 2005

Calculated measures can be used in filters or the where clause.

Syntax for calculated measure expressions

The syntax for a calculated measure consists of calculations embedded in the tags <EXPRESSION></EXPRESSION>.

Universe design tool functions are allowed in calculated measure expressions, such as:

- @Select
- @Prompt
- @Variable
- @Where

> **Note**
>
> Expressions for calculated measures cannot include the @Aggregate_Aware function. The check integrity function validates the XML syntax and any of the above listed @Functions, including those inserted in the MDX statements. However, the MDX statements are not parsed.

Constants are allowed in the expression, for example "10" or "ABC".

Calculated measures can reference any OLAP metadata:

- measures
- dimensions
- dimension levels
- MDX expressions

Recommendations for calculated measure expressions

Use @Select (Measure name) rather than the measure definition for the following reasons:

- @Select is resolved at query time.
- Calculated measures can reference other calculated measures if they are inside a @Select function.
- The validity of objects inside the @Select function is checked.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Generate and set the index awareness for each object definition.

Use a reference to an object or detail whose definition refers to the Technical Name or Unique Name of the level or attribute.

Example

Calculated measure expression

```
<EXPRESSION>@Select(Key Figures\Order Amount)*@Select(Key Figures\Order
Quantity)</EXPRESSION>
```

Related Information

To create a calculated measure in an OLAP universe [page 35]

7.8.1 To create a calculated measure in an OLAP universe

To create a calculated measure in an OLAP universe:

1. In the universe design tool, open an OLAP universe.
2. Insert a new measure object in the universe.
3. In the *Where:* box, enter or paste the object definition as an XML/MDX expression.
4. Click *Parse* to review the object definition and fix any errors.
5. Click *OK* to save the object definition.
6. Select *Tools* > *Check Integrity*.
   The integrity check validates the XML syntax and any universe design tool @FUNCTIONS.

Related Information

Calculated measures in OLAP universes [page 34]

7.9 About MDX functions for cube queries

Use the MDX editor to define your cube queries.

When adding a new object or a predefined filter to an OLAP universe, there is a list of supported MDX expressions corresponding to the specific data source connection.

A library of available expressions is stored in the `.prm` connection file. When you open the Edit Properties pane for an object and opend the Edit Select pane for the query, the available expressions are displayed in the Functions

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

pane. To insert the expression in the SELECT or WHERE statement, click in the position in the expression where you want to insert the expression and double-click on the appropriate expression.

OLAP Universe MDX Dictionary - List of Functions (PRM file)

When adding a new object or a predefined filter to an OLAP universe, an explicit list of MDX functions (mainly member functions) and operators is made available in the object and filter editors for the appropriate OLAP connection (SAP or MSAS) that you can use in the expression. For a description of how to set the connectivity for SAP or mySQL (sap.prm, sqlsrv_as.prm), refer to the Data Access Guide. The available functions and operators depend on the connection type of the universe. This list of functions is provided by the PRM file for each connectivity. It does not provide the whole list of supported functions, only the most frequently used functions.

The following MDX operators are available for queries.

- Equal
- NotEqual
- InList
- NotInList
- Greater
- GreaterOrEqual
- Less
- LessOrEqual
- Between
- NotBetween
- Like
- NotLike

The list below shows examples of some of the available MDX folder functions when editing a condition. The available functions depend on the underlying database.

- Set functions (ADDCALCULATEDMEMBERS, ALLMEMBERS ...)
- Statistical/Numeric functions (AGGREGATE, AVG ...)
- Navigation/Member functions (ANCESTOR, ASCENDANTS...)
- Metadata functions (AXIS, HIERARCHY...)

7.10 XML syntax for filter and WHERE statements

This section describes the XML syntax for defining the WHERE clause or filter statements in your OLAP universe. You must add the FILTER or FILTER EXPRESSION tags manually, and then enter your expression between the tags either manually or with the universe design tool MDX editor.

- Use `<FILTER= "your_object_definition">` when using a single object definition. Enter your object definition inside the double quotes.
- Use `<FILTER EXPRESSION= "yourcomplexMDX_expression">` when using a complex MDX expression containing one or more objects. Enter your expression inside the double quotes.

The syntax for a single filter object is as follows:

```
<FILTER = “your_object_definition”><CONDITION
OPERATORCONDITION="yourOperator"><CONSTANT VALUE="your_Value"/></CONDITION></FILTER>
```


© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Where:

- yourMDX_expression is the single object definition, enclosed in double quotes.
- `CONSTANT VALUE` is either `CONSTANT CAPTION` or `CONSTANT TECH_NAME`
- `yourOperator` is the filter expression operator (`equals`, `inlist`...). When the `InIist` operator is used, you must insert a `CONSTANT CAPTION` or `CONSTANT TECH_NAME` element for each item in the list.
- `your_Value` is the defined filter value when `CONSTANT CAPTION` is used, or the object identifier if `CONSTANT TECH_NAME` is used.

The syntax for a single filter object using the `InList` operator, where three countries are listed, is as follows:

```
<FILTER= "your_object_definition "><CONDITION OPERATORCONDITION="InList"><CONSTANT
CAPTION="England"/><CONSTANT CAPTION="France"/><CONSTANT CAPTION="Germany"/></
CONDITION></FILTER>
```

The syntax for a complex filter expression and the TECH_NAME for the filtered value is as follows:

```
<FILTER EXPRESSION="yourComplex_MDX_Expression"><CONDITION
OPERATORCONDITION="Equal"><CONSTANT TECH_NAME="1"/></CONDITION></FILTER>
```

Example

Filter with a calculated member in the filter expression

```
<FILTER EXPRESSION="IIF([0CALYEAR].CurrentMember > “2000”, 1,0)"><CONDITION
OPERATORCONDITION="Equal"><CONSTANT CAPTION="1"/></CONDITION></FILTER>
```

7.11 Predefined conditions in OLAP universes

Predefined conditions in OLAP universes are like conditions in non-OLAP universes except that you define the WHERE clause using XML rather than SQL. You can declare filters manually, or by using the Predefined filter editor.

7.11.1 XML Syntax for predefined filter options

Syntax for predefined conditions

A single predefined condition may contain multiple filters combined with the AND and OR operators. By default, all filters are combined with the AND operator. To include filters using OR, you must use the AND and OR operator tags.

The functions @Select, @Prompt and @Variable are allowed in the predefined filter definition.

Predefined filters can include one or multiple prompts. Prompts can be mandatory or optional.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Example

Using AND and OR tags for pre-defined conditions

```
<OPERATOR VALUE="AND">
    <FILTER "[Level Object definition]">
        <CONDITION OPERATORCONDITION="Operator">
           <CONSTANT Level Attribute="Value"/>
        </CONDITION>
    </FILTER>
    <OPERATOR VALUE="OR">
        <FILTER "[Level Object definition]">
            <CONDITION OPERATORCONDITION="Operator">
                <CONSTANT Level Attribute="Value"/>
            </CONDITION>
        </FILTER>
        <FILTER "[Level Object definition]">
            <CONDITION OPERATORCONDITION="Operator">
                <CONSTANT Level Attribute="Value"/>
                </CONDITION>
            </FILTER>
    </OPERATOR>
</OPERATOR>
```

7.11.2 Manually creating pre-defined conditions in an OLAP universe

To create a pre-defined condition:

1. In universe design tool, open an OLAP universe and click the conditions radio button at the bottom of the Universe pane.
   The conditions view of the Universe pane appears. It contains a tree view of the classes in the universe.
2. Right-click a class and select *Condition...* from the contextual menu.
3. In the *Where:* box, edit the XML template filter.

   The template filter has the format:

   ```
   <FILTER "[Level Object definition]">
       <CONDITION OPERATORCONDITION="Operator">
           <CONSTANT Level Attribute="Value"/
       </CONDITION>
   </FILTER>
   ```

   Replace the elements in the template as follows:

Template element:	Possible Values:
Level Object definition	Enter the dimension level or measure implied in the filter. Enter the object definition not the object name.
Operator	Enter one of: ○ Equal

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Template element:	Possible Values:
	○ NotEqual ○ Greater ○ Less ○ GreaterOrEqual ○ LessOrEqual ○ Between ○ NotBetween ○ InList ○ NotInList ○ Like ○ NotLike
Level Attribute	Enter one of: ○ NAME ○ CAPTION ○ TECH_NAME ○ DESCRIPTION
Value	Enter the value or a prompt. Define one value per CONSTANT tag.

An example of an edited predefined condition:

```
<FILTER KEY="[0D_DIV].[LEVEL01]">
    <CONDITION OPERATORCONDITION="InList">
        <CONSTANT CAPTION="Internal"/>
        <CONSTANT CAPTION="Service"/>
    </CONDITION>
</FILTER>
```

4. Click *Parse* to review the syntax and fix any errors.
5. Click *OK* to save the condition.

Related Information

Predefined conditions in OLAP universes [page 37]
Optional prompts in OLAP universes [page 42]

7.11.3 About the Pre-defined Filter editor

The *Pre-defined filter* editor is for editing pre-defined filters in OLAP universes. Use it to select objects, operators, lists of values, prompts, functions, and other optional elements that can be used to define a filter for your OLAP universe.

In the condition properties panel of a filter, you can manually type the filter expression or click >> to open the *Pre-defined filter* editor. When the editor is open, you can insert an `@Prompt` in the filter expression: Right-click at the

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

appropriate point in the filter expression and select *New @Prompt* from the shortcut menu. The pre-defined filter editor inserts the filter expression in the query/object definition.

Example

Restriction on dimension Customer at the country level to restrict country to Canada

<FILTER KEY="[Customer].[Country].[Country]"> <CONDITION OPERATORCONDITION="Equal">
<CONSTANT CAPTION="Canada" /> </CONDITION> </FILTER>

Related Information

About the options for the Pre-defined Filter editor [page 40]
Editing a pre-defined filter with the pre-defined filter editor [page 41]
About MDX functions for cube queries [page 35]

7.11.4 About the options for the Pre-defined Filter editor

The *Pre-defined Filter* editor allows you to easily define a universe filter for an OLAP universe. You can define the following options:

Option	Description
Select an Operator	Select an operator from the available list. Default = *Equal*
Base the filter on	Filter either on an existing universe object or on a free definition (for example: [Measures].[Internet Sales Amount]). Default = *Universe object*.
Select a LoV	Select a list of objects in the current universe when filter based on existing object. Default selection = The Root class in the list of objects.
Comparison values	Define values to compare the object/expression to. Depending on selected operator, there are one or two sets of values to enter. Values can either be static or based on a prompt. Default = *Static values*.
Insert a Prompt	Edit a prompt manually, or use the *@Prompt* editor. Click >> to open the *@Prompt* editor.
Set Index Awareness	Enable the index awareness function. Primary keys must be declared for this to work correctly. When index awareness is set up in the universe design tool, the primary and foreign key columns are used to speed up data retrieval and to allow the universe design tool to generate more efficient SQL filters. Default = Not selected.
Use calculated expression	When selected, this encloses the filter expression inside <EXPRESSION> </EXPRESSION> tags. Default = Not selected.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Option	Description
Optional	Set the current filter expression as optional. This applies to the current filter expression in the filter editor only and not to the entire pre-defined condition object. Default = Not selected.

> **Note**
>
> "Optional" tags cannot be used for pre-defined filters in Web Intelligence. If these tags are used, they are treated as a mandatory part of the query, therefore preventing the query from running.

Related Information

Editing a pre-defined filter with the pre-defined filter editor [page 41]

7.11.5 Editing a pre-defined filter with the pre-defined filter editor

You are editing a filter in an OLAP universe.

The *Pre-defined Filter* editor is updated as you select or enter the values. You can right-click in the filter expression to insert an `@Prompt` expression in the filter expression. When you right-click and select *New @Prompt*, the *Prompt* editor opens.

1. In the *Properties* pane of the condition (filter) pane, click >>.
   The *Pre-defined Filter* editor appears.
2. To base the filter on a universe object, select *Universe object* and choose an object from the *Available Objects* pane. To base the predefined filter on your own expression, select *Free definition* and type the expression in the *Available Objects* pane.
3. Select an operator from the *Operators* list. Multiple values (right operand) are allowed for the In List and Not In List operators only.
4. Select *Static value* to define one or more fixed values, or select *Prompt* to insert a prompt expression.
   When you select *Prompt* , the *Edit* button is activated. Click *Edit* to open the *@Prompt* editor and define the prompt expression as required.
5. Click *OK* to validate the filter definition.
   The parser checks the syntax for errors, including the integrity check. If errors are found a warning message with error message is shown. When no errors are found, the new condition object is added to the universe with the filter definition.

Related Information

About the options for the Pre-defined Filter editor [page 40]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

About the Pre-defined Filter editor [page 39]

7.12 Optional prompts in OLAP universes

Universes generated from OLAP data sources support optional prompts.

For SAP BW optional variables, filters with the optional conditions are automatically generated in the universe.

In pre-defined conditions or in an object's WHERE clause, to make a prompt optional, embed the XML filter expression between the two XML tags: <OPTIONAL> and </OPTIONAL>.

Example

Optional prompt in a pre-defined condition

```
<OPTIONAL>
    <FILTER KEY="[Products].[Family]" >
        <CONDITION OPERATORCONDITION="InList" >
            <CONSTANT CAPTION="@prompt('Enter value(s) for Product
family:','A','Products\Family',Multi,primary_key,persistent)"/>
        </CONDITION>
     </FILTER>
</OPTIONAL>
```

Related Information

Manually creating pre-defined conditions in an OLAP universe [page 38]

7.13 To improve performance of certain queries on SAP BW universes

For queries on SAP BW universes that include only the key and medium name detail objects of a dimension, you can modify the generated syntax of the objects to improve query performance.

To modify the syntax:

1. Open the universe in universe design tool.
2. Double click the key detail object you want to modify.
3. In the Select text box on the *Definition* tab of the *Edit Properties* dialog box, change the syntax to refer to the NAME attribute of the SAP characteristic.
   For example, for the object *L01 Customer Key*, change the generated select syntax:

   ```
   [Z_CUSTOM].[LEVEL01].[[2Z_CUSTOM]].[Value]
   ```


© 2015 SAP SE or an SAP affiliate company. All rights reserved.

to refer to the NAME attribute:

```
[Z_CUSTOM].[LEVEL01].[NAME]
```

4. Click *OK* to save the changes.
5. Follow the same steps for the name object. Change the syntax to refer to the DESCRIPTION attribute of the SAP characteristic.
   For example, for the object *L01 Customer Medium Name*, change the generated select syntax:

   ```
   [Z_CUSTOM].[LEVEL01].[[5Z_CUSTOM]].[Value]
   ```

   to refer to the DESCRIPTION attribute:

   ```
   [Z_CUSTOM].[LEVEL01].[DESCRIPTION]
   ```


© 2015 SAP SE or an SAP affiliate company. All rights reserved.

8 OLAP universe lifecycle management

8.1 About OLAP universe lifecycle management

> **Note**
>
> When you open a universe created with a version of Universe Designer prior to XIR3.1 SP2, you must refresh and save the universe before making any changes in the universe or in the OLAP source.

OLAP universes are generated automatically from an OLAP data source (for example, an SAP BEx Query or an MSAS 2005 cube). In the universe design tool, you can create and change objects in the existing OLAP universe.

The *Update OLAP Universe* wizard allows you to refresh the structure of OLAP universes automatically with changes made in the OLAP data source. The wizard compares the universe with the updated data source. The wizard can differentiate generated objects from objects added or modified manually, allowing you to keep the changes made manually in the universe design tool. The wizard does not update objects that were manually added in the universe design tool.

What can be detected and updated depends on the items and the data source, as shown in the table below.

What the wizard can detect	New items can be detected in	Modified items can be detected in	Deleted items can be detected in
Dimensions	All data sources	All data sources	All data sources
Hierarchies	SAP BW and MSAS only	All data sources	All data sources
Levels	All data sources	All data sources	All data sources
Properties	MSAS only	MSAS only	MSAS only
Measures	All data sources	All data sources	All data sources
SAP BW Variables	SAP BW only	SAP BW only	SAP BW only
Sub-classes	All data sources	All data sources	All data sources

> **Note**
>
> When you update a universe created with a version of Universe Designer earlier than XIR3.1 SP2, if the name of a dimension has changed in the SAP cube, the refresh of the dimension does not work: The dimension is duplicated in the universe. You must manually update the classes in the universe.

Related Information

To refresh an OLAP universe [page 47]
Overview: relationship between universe object status and OLAP object status [page 45]
How dimensions are managed in OLAP universe updates [page 50]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

How hierarchies or characteristics are managed in OLAP universe updates [page 55]
How levels are managed in OLAP universe updates [page 62]
How SAP variables are managed in OLAP universe updates [page 65]
How key figures or measures are managed in OLAP universe updates [page 68]
How SAP key dates are managed in OLAP universe updates [page 72]

8.2 Overview: relationship between universe object status and OLAP object status

The table below gives a brief overview of the relationship between an SAP OLAP object state and universe object state. There are specific notes concerning most of the actions, for more information refer to the more detailed sections in this chapter.

Table 9:

OLAP metadata		Universe object status				
		Unchanged	Updated*	Deleted	Moved	Hidden
Dimension		Universe equivalent = class				
Status	Unchanged	NoC	Upd	NoC	NoC	NoC
	Updated*	Upd	Upd	NoC	Upd	Upd
	Deleted	Del/Ob	Del/Ob	NoC	Del/Ob	NoC
	Moved	Move	NoC	NoC	NoC	Move
	Characteristic created	CreS	CreS	N/A	CreS	CreS
	Created	Cre	Cre	N/A	Cre	Cre
Hierarchy or Characteristic		Universe equivalent = sub-class				
Status	Unchanged	NoC	Upd	NoC	NoC	NoC
	Updated*	Upd	Upd	NoC	Upd	Upd
	Changed	UpdMDX	UpdMDX	NoC	UpdMDX	UpdMDX
	Display Attribute	Cre	Cre	Cre	Cre	Cre
	Navigation Attribute	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	Deleted	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	Moved	Move	Move	NoC	Move	Move
	New	Cre	Cre	Cre	Cre	Cre
Level		Universe equivalent = level				
Status	Unchanged	NoC	NoC	NoC	NoC	NoC
	Updated*	Upd	Upd	NoC	Upd	Upd

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

OLAP metadata		Universe object status				
		Unchanged	Updated*	Deleted	Moved	Hidden
	Deleted	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	Moved	Move	Move	NoC	Move	Move
	New	Cre	Cre	Cre	Cre	Cre
Variable		Universe equivalent = filter				
Status	Unchanged	NoC	NoC	NoC	NoC	NoC
	Updated*	Upd	Upd	Cre	Upd	Upd
	Deleted	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	New	Cre	Cre	Cre	Cre	Cre
Key figure		Universe equivalent = measure				
Status	Unchanged	NoC	NoC	NoC	NoC	NoC
	Updated*	Upd	Upd	NoC	Upd	Upd
	Deleted	Del/Ob	Del/Ob	NoC	Del/Ob	Del/Ob
	Moved	Move	Move	NoC	Move	Move
	New	Cre	Cre	Cre	Cre	Cre
Key date		Universe equivalent = parameter				
Status	Unchanged	NoC	N/A	Cre	N/A	N/A
	Deleted	Del	N/A	N/A	N/A	N/A
	New	Cre	N/A	Cre	N/A	N/A

LEGEND:

- *:The one of the object properties (name, description...) has changed.
- Cre: Create the equivalent object
- CreS: Create the equivalent sub-class object
- Del/Ob: Deleted or Obsolete (obsolete objects are hidden and their names prefixed with ##)
- Move: The object is moved
- N/A: Does not apply
- NoC: No change
- Upd: Updated
- UpdMDX: Update the MDX definition

Related Information

To refresh an OLAP universe [page 47]
How dimensions are managed in OLAP universe updates [page 50]
How hierarchies or characteristics are managed in OLAP universe updates [page 55]
How levels are managed in OLAP universe updates [page 62]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

How SAP variables are managed in OLAP universe updates [page 65]
How key figures or measures are managed in OLAP universe updates [page 68]
How SAP key dates are managed in OLAP universe updates [page 72]

8.3 To refresh an OLAP universe

To refresh the structure of an OLAP universe:

- In the universe design tool, open the universe you want to update.
- Select *View* > *Refresh Structure*.
  The Update OLAP Universe wizard appears.
- Click *Begin*.

  The Updated metadata definitions page appears.

  Select keep options when you want to preserve modifications to objects in the universe that were made manually. All keep options are selected by default. You can choose to keep the following attributes:

Option	Description
Keep business name	Class, Dimension, Measure, Detail and Condition names.
Keep type of objects	If the object has changed in the universe (for example, a detail is changed to a dimension, the update will not reintroduce the initial object type). This concerns Dimensions, Measures, and Details.
Keep description of objects	When this is selected, if the description is updated in the OLAP source, the universe is not updated with this information.
Keep data type of objects	Character, Numeric, Date, and Long text objects.
Keep list of values options of objects	You can keep the options that were initially set: ○ Associate a list of values ○ Automatic refresh before use ○ Hierarchical display ○ Export with universe ○ Delegate search
Keep advanced options of objects	The options are: Security access level The object can be used in ○ Used in result ○ Used in condition ○ Used in sort
Delete obsolete objects	Items that are no longer in the data source will be deleted from the universe.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Option	Description
Hide obsolete objects	Items that are no longer in the cube will be hidden in the universe and prefixed by /##/.

- Select the desired options and click *Next*.

  The *Change Management Results* page appears showing added/deleted/hidden objects. Hidden objects are moved to a separate class in the universe and appear in italic font preceded by /##/.
- In the Added metadata options pane, set the options for how added metadata is treated

Generic OLAP options	Description
Generate technical names as details	You can set the application to generate the technical name of the universe as a property. When the universe is generated, this creates an object that points to the technical name.
Regenerate all manually deleted objects	Any manually deleted universe objects will be regenerated.

SAP OLAP options	Description
Set measures aggregation to delegated	You can set the application to set the aggregation function of measures to database delegated.
Replace prefixes L00, L01,	Universe level prefixes indicate the level in the hierarchy of an object. Level L00 is the top or root level, L01 is the next level down. In the "New Universe Wizard", you can replace OLAP universe level prefixes with a different prefix. The level numbering is retained, but the prefix 'L' can be replaced by Level, for example. Type your own prefix in the *New prefix* field. This prefix is prepended to all levels throughout the OLAP universe.
Rename level 00 to All	This option is disabled if *Generate level 00* is set to *No*. You can rename the top level (root level) L00 to All the next time the universe is generated.
Generate level 00	This option only applies to SAP Characteristics. You can disactivate this option for Characteristics only. Level 00 is always generated for hierarchies and hierarchy variables. You can regenerate the Level numbers (L00, L01, L02...) when you generate or update a universe. The level numbers are appended to the level names (e.g. "Monthly Sales_L01"). This is useful for Web Intelligence reports, where the All level is used to aggregate results for a query. This avoids having to create the aggregate field in the Web Intelligence report.

- On the Change Management results page, select one of:

Option	Description
OK	If you are not satisfied with the results, you can click *OK*, then close the Universe without saving or exporting.
Export	If you are satisfied with the changes, click *Export* to save and export the updated universe to the CMS.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Option	Description
Check Integrity	Click *Check Integrity* to perform an integrity check. It checks the structure, parses objects, parses joins, parses conditions, and checks cardinality. When the check is complete, the *Integrity Check Results* page appears. From this page you can print the results of the check.

If you do not see all changes to the universe that you expect, stop, then restart the universe design tool before trying the update again. This makes a new connection to the data source and clears the cache.

Related Information

Synchronizing the universe and the OLAP cube [page 50]
About OLAP universe lifecycle management [page 44]

8.4 Regenerating Level 00 for OLAP universes

You can regenerate the Level numbers (L00, L01, L02...) when you generate or update a universe. The level numbers are appended to the level names (e.g. "Monthly Sales_L01")

Related Information

Setting delegated measures for OLAP universes [page 32]
Replacing OLAP universe level prefixes [page 50]
Renaming level L00 to ALL [page 49]

8.5 Renaming level L00 to ALL

You can rename the top level (root level) L00 to ALL when the universe is next generated. This is useful for SAP BusinessObjects Web Intelligence reports, where the ALL level is used to aggregate results for a query. This avoids having to create the aggregate field in the Web Intelligence report.

Related Information

Setting delegated measures for OLAP universes [page 32]
Replacing OLAP universe level prefixes [page 50]
Regenerating Level 00 for OLAP universes [page 49]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

8.6 Replacing OLAP universe level prefixes

Universe level prefixes indicate the level in the hierarchy of an object. Level L00 is the top or root level, L01 is the next level down. In the *New Universe Wizard*, you can replace OLAP universe level prefixes with a different prefix. The level numbering is retained, but the prefix 'L' can be replaced by Level, for example. Type your own prefix in the *New prefix* field. This prefix is prepended to all levels throughout the OLAP universe.

Related Information

Setting delegated measures for OLAP universes [page 32]
Regenerating Level 00 for OLAP universes [page 49]
Renaming level L00 to ALL [page 49]

8.7 Synchronizing the universe and the OLAP cube

When you update a universe, the objects in the universe are compared with the objects in the OLAP cube. The comparison ensures that changes made in the cube do not adversely affect the universe. This means that any objects used (and even deleted) in the universe must always be available. Any new objects in the OLAP cube are made available to the universe. To see how the different objects are affected by changes, see the links below.

When object properties are updated, only certain properties are updated in the universe, other properties should not change. The following table shows what happens.

Related Information

To refresh an OLAP universe [page 47]
How dimensions are managed in OLAP universe updates [page 50]
How hierarchies or characteristics are managed in OLAP universe updates [page 55]
How levels are managed in OLAP universe updates [page 62]
How SAP variables are managed in OLAP universe updates [page 65]
How key figures or measures are managed in OLAP universe updates [page 68]
How SAP key dates are managed in OLAP universe updates [page 72]

8.8 How dimensions are managed in OLAP universe updates

This applies to SAP, MSAS, and Essbase data sources. The universe class is the equivalent of the OLAP dimension. How the universe objects are managed with respect to the OLAP objects depends on the type of

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

change. Refer to the topics listed below to see how the universe objects are impacted by specific OLAP object changes.

Related Information

To refresh an OLAP universe [page 47]
When a dimension is unchanged [page 51]
When a dimension is updated (name, description) [page 52]
When a dimension is deleted [page 52]
When a dimension is moved [page 53]
When a hierachy or characteristic is created [page 54]
When a dimension is new [page 54]

8.8.1 When a dimension is unchanged

The following table shows what happens to the equivalent universe class in the different possible situations when the dimension is unchanged:

Table 10:

When the universe class	What happens to the universe class
Is unchanged	The universe class is unchanged
Is updated	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.
Is deleted	No change to the universe class. Create the object if the option *Regenerate Objects manually deleted* is selected. Do not regenerate children that had not been deleted
Is moved	No change to the universe class.
Is hidden	No change to the universe class.

Related Information

To refresh an OLAP universe [page 47]
When a dimension is updated (name, description) [page 52]
When a dimension is deleted [page 52]
When a dimension is moved [page 53]
When a hierachy or characteristic is created [page 54]
When a dimension is new [page 54]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

8.8.2 When a dimension is updated (name, description)

The following table shows what happens to the equivalent universe class in the different possible situations when the name or description of the dimension is updated:

Table 11:

When the universe class	What happens to the universe class
Is unchanged	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.
Is updated	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.
Is deleted	No change to the universe class. Create if the option *Regenerate Objects manually deleted* is selected. Do not regenerate children that had not been deleted.
Is moved	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.
Is hidden	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.

Related Information

To refresh an OLAP universe [page 47]
When a dimension is unchanged [page 51]
When a dimension is deleted [page 52]
When a dimension is moved [page 53]
When a hierachy or characteristic is created [page 54]
When a dimension is new [page 54]

8.8.3 When a dimension is deleted

The following table shows what happens to the equivalent universe class in the different possible situations when the dimension is deleted:

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Table 12:

When the universe class	What happens to the universe class
Is unchanged	Delete if the option *Delete obsolete objects* is selected. Make the sub-class obsolete if the option *Hide obsolete objects* is selected. Do not delete if the class contains custom objects
Is updated	Delete if the option *Delete obsolete objects* is selected. Make the sub-class obsolete if the option *Hide obsolete objects* is selected. Do not delete if the class contains custom objects
Is deleted	No change to the universe class.
Is moved	Delete if the option *Delete obsolete objects* is selected. Make the sub-class obsolete if the option *Hide obsolete objects* is selected. Do not delete if the class contains custom objects
Is hidden	No change to the universe class

Related Information

To refresh an OLAP universe [page 47]
When a dimension is unchanged [page 51]
When a dimension is updated (name, description) [page 52]
When a dimension is moved [page 53]
When a hierachy or characteristic is created [page 54]
When a dimension is new [page 54]

8.8.4 When a dimension is moved

The following table shows what happens to the equivalent universe class in the different possible situations when the dimension is moved:

Table 13:

When the universe class	What happens to the universe class
Is unchanged	Move the class accordingly
Is updated	No change
Is deleted	No change. Create if the option *Regenerate Objects manually deleted* = *Yes* Do not regenerate children that had not been deleted
Is moved	No change
Is hidden	Move the class accordingly

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Related Information

To refresh an OLAP universe [page 47]
When a dimension is unchanged [page 51]
When a dimension is updated (name, description) [page 52]
When a dimension is deleted [page 52]
When a hierachy or characteristic is created [page 54]
When a dimension is new [page 54]

8.8.5 When a hierachy or characteristic is created

Hierachy applies to MSAS or Essbase data sources, characteristic applies to SAP data sources. The following table shows what happens to the equivalent universe class in the different possible situations when the SAP characteristic is created:

Table 14:

When the universe class	What happens to the universe class
Is unchanged	Create sub-class
Is updated	Create sub-class
Is deleted	Does not apply
Is moved	Create sub-class
Is hidden	Create sub-class

Related Information

To refresh an OLAP universe [page 47]
When a dimension is unchanged [page 51]
When a dimension is updated (name, description) [page 52]
When a dimension is deleted [page 52]
When a dimension is moved [page 53]
When a dimension is new [page 54]

8.8.6 When a dimension is new

The universe class is created when the dimension is created.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Related Information

To refresh an OLAP universe [page 47]
When a dimension is unchanged [page 51]
When a dimension is updated (name, description) [page 52]
When a dimension is deleted [page 52]
When a dimension is moved [page 53]
When a hierachy or characteristic is created [page 54]

8.9 How hierarchies or characteristics are managed in OLAP universe updates

This section applies to hierarchies for MSAS and Essbase data sources, and characteristics for SAP data sources. The universe sub-class is the equivalent of the OLAP characteristic. How the universe objects are managed with respect to the OLAP objects depends on the type of change. Refer to the topics listed below to see how the universe objects are impacted by specific OLAP object changes.

Related Information

To refresh an OLAP universe [page 47]
When a hierarchy or characteristic is unchanged [page 55]
When a characteristic business name or description is updated [page 56]
When a characteristic's active hierarchy is changed [page 57]
When a characteristic display attribute is changed to a navigation attribute [page 58]
When a hierarchy or characteristic is deleted [page 60]
When a hierarchy or characteristic is moved [page 60]
When a hierarchy or characteristic is new [page 61]

8.9.1 When a hierarchy or characteristic is unchanged

The following table shows what happens to the equivalent universe sub-class in the different possible situations when the hierarchy or characteristic is unchanged:

Table 15:

When the universe sub-class	What happens to the universe sub-class
Is unchanged	No change
Is updated	No change

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

When the universe sub-class	What happens to the universe sub-class
Is deleted	No change. Create if Option: *Regenerate Objects manually deleted* = *Yes* Do not regenerate children levels that had not been deleted
Is moved	No change
Is hidden	No change

Related Information

To refresh an OLAP universe [page 47]
When a characteristic business name or description is updated [page 56]
When a characteristic's active hierarchy is changed [page 57]
When a characteristic display attribute is changed to a navigation attribute [page 58]
When a hierarchy or characteristic is deleted [page 60]
When a hierarchy or characteristic is moved [page 60]
When a hierarchy or characteristic is new [page 61]

8.9.2 When a characteristic business name or description is updated

The following table shows what happens to the equivalent universe sub-class in the different possible situations when the characteristic is updated:

Table 16:

When the universe sub-class	What happens to the universe sub-class
Is unchanged	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.
Is updated	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.
Is deleted	No change. Create if Option: *Regenerate Objects manually deleted* = *Yes* Do not regenerate children levels that had not been deleted

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

When the universe sub-class	What happens to the universe sub-class
Is moved	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are not selected.
Is hidden	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.

Related Information

To refresh an OLAP universe [page 47]
When a hierarchy or characteristic is unchanged [page 55]
When a characteristic's active hierarchy is changed [page 57]
When a characteristic display attribute is changed to a navigation attribute [page 58]
When a hierarchy or characteristic is deleted [page 60]
When a hierarchy or characteristic is moved [page 60]
When a hierarchy or characteristic is new [page 61]

8.9.3 When a characteristic's active hierarchy is changed

This applies to SAP data sources only. The following table shows what happens to the equivalent universe sub-class in the different possible situations when the characteristic's active hierarchy has changed:

Table 17:

When the universe sub-class	What happens to the universe sub-class
Is unchanged	Update the MDX definitions of existing objects in the sub-class to reference the new active hierarchy. Reports built before the refresh continue to work
Is updated	Update the MDX definitions of existing objects in the sub-class to reference the new active hierarchy. Reports built before the refresh continue to work
Is deleted	No change. Create if the option *Regenerate Objects manually deleted* = *Yes* Does not regenerate children levels that were not deleted.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

When the universe sub-class	What happens to the universe sub-class
Is moved	Update the MDX definitions of existing objects in the sub-class to reference the new active hierarchy. Reports built before the refresh continue to work.
Is hidden	Update the MDX definitions of existing objects in the sub-class to reference the new active hierarchy.

Related Information

To refresh an OLAP universe [page 47]
When a hierarchy or characteristic is unchanged [page 55]
When a characteristic business name or description is updated [page 56]
When a characteristic display attribute is changed to a navigation attribute [page 58]
When a hierarchy or characteristic is deleted [page 60]
When a hierarchy or characteristic is moved [page 60]
When a hierarchy or characteristic is new [page 61]

8.9.4 When a characteristic display attribute is changed to a navigation attribute

This applies to SAP data sources only. The following table shows what happens to the equivalent universe sub-class in the different possible situations when the characteristic display attribute is changed to a navigation attribute:

Table 18:

When the universe sub-class	What happens to the universe sub-class
Is unchanged	Create
Is updated	Create
Is deleted	Create
Is moved	Create
Is hidden	Create

Related Information

To refresh an OLAP universe [page 47]
When a hierarchy or characteristic is unchanged [page 55]
When a characteristic business name or description is updated [page 56]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

When a characteristic's active hierarchy is changed [page 57]
When a hierarchy or characteristic is deleted [page 60]
When a hierarchy or characteristic is moved [page 60]
When a hierarchy or characteristic is new [page 61]

8.9.5 When a characteristic navigation attribute is changed to a display attribute

This applies to SAP data sources only. The following table shows what happens to the equivalent universe sub-class in the different possible situations when the hierarchy or characteristic navigation attribute is changed to a display attribute:

Table 19:

When the universe sub-class	What happens to the universe sub-class
Is unchanged	Delete if the option *Delete obsolete objects* is selected. Make the sub-class hidden if the option *Hide obsolete objects* is selected. Does not delete if the class contains custom objects.
Is updated	Delete if the option *Delete obsolete objects* is selected. Make the sub-class hidden if the option *Hide obsolete objects* is selected. Does not delete if the class contains custom objects.
Is deleted	No change
Is moved	Delete if the option *Delete obsolete objects* is selected. Make the sub-class hidden if the option *Hide obsolete objects* is selected. Does not delete if the class contains custom objects.
Is hidden	Delete if the option *Delete obsolete objects* is selected. Make the sub-class hidden if the option *Hide obsolete objects* is selected. Does not delete if the class contains custom objects.

Related Information

To refresh an OLAP universe [page 47]
When a hierarchy or characteristic is unchanged [page 55]
When a characteristic business name or description is updated [page 56]
When a characteristic's active hierarchy is changed [page 57]
When a hierarchy or characteristic is deleted [page 60]
When a hierarchy or characteristic is moved [page 60]
When a hierarchy or characteristic is new [page 61]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

8.9.6 When a hierarchy or characteristic is deleted

The following table shows what happens to the equivalent universe sub-class in the different possible situations when the hierarchy or characteristic is deleted:

Table 20:

When the universe sub-class	What happens to the universe sub-class
Is unchanged	Delete if the option *Delete obsolete objects* is selected. Make the sub-class obsolete if the option *Hide obsolete objects* is selected. Do not delete if the sub-class contains custom objects.
Is updated	Delete if the option *Delete obsolete objects* is selected, make the sub-class obsolete if the option *Hide obsolete objects* is selected. Do not delete if the sub-class contains custom objects.
Is deleted	No change
Is moved	Delete if the option *Delete obsolete objects* is selected, make the sub-class obsolete if the option *Hide obsolete objects* is selected. Do not delete if the sub-class contains custom objects.
Is hidden	Delete if the option *Delete obsolete objects* is selected, make the sub-class obsolete if the option *Hide obsolete objects* is selected. Do not delete if the sub-class contains custom objects.

Related Information

To refresh an OLAP universe [page 47]
When a hierarchy or characteristic is unchanged [page 55]
When a characteristic business name or description is updated [page 56]
When a characteristic's active hierarchy is changed [page 57]
When a characteristic display attribute is changed to a navigation attribute [page 58]
When a hierarchy or characteristic is moved [page 60]
When a hierarchy or characteristic is new [page 61]

8.9.7 When a hierarchy or characteristic is moved

If the characteristic is moved within the same dimension, there is no change: ignore the table below. The following table shows what happens to the equivalent universe sub-class in the different possible situations when the hierarchy or characteristic is moved to another dimension:

Table 21:

When the universe sub-class	What happens to the universe sub-class
Is unchanged	Move the sub-class accordingly.
Is updated	Move the sub-class accordingly.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

When the universe sub-class	What happens to the universe sub-class
Is deleted	No change. Create if the option: *Regenerate Objects manually deleted* = *Yes* Do not regenerate children levels that had not been deleted.
Is moved	No change.
Is hidden	Move the sub-class accordingly.

Related Information

To refresh an OLAP universe [page 47]
When a hierarchy or characteristic is unchanged [page 55]
When a characteristic business name or description is updated [page 56]
When a characteristic's active hierarchy is changed [page 57]
When a characteristic display attribute is changed to a navigation attribute [page 58]
When a hierarchy or characteristic is deleted [page 60]
When a hierarchy or characteristic is new [page 61]

8.9.8 When a hierarchy or characteristic is new

The universe sub-class created when the hierarchy or characteristic is created.

Related Information

To refresh an OLAP universe [page 47]
When a hierarchy or characteristic is unchanged [page 55]
When a characteristic business name or description is updated [page 56]
When a characteristic's active hierarchy is changed [page 57]
When a characteristic display attribute is changed to a navigation attribute [page 58]
When a hierarchy or characteristic is deleted [page 60]
When a hierarchy or characteristic is moved [page 60]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

8.10 How levels are managed in OLAP universe updates

> **Note**
>
> In the universe, do not move levels to other hierarchies. If you want to move a level, copy and paste the level to the new hierarchy.

The universe level or dimension object is the equivalent of the OLAP level. How the universe objects are managed with respect to the OLAP objects depends on the type of change. Refer to the topics listed below to see how the universe objects are impacted by specific OLAP object changes.

Related Information

To refresh an OLAP universe [page 47]
When a level is unchanged [page 62]
When the name or description of a level is updated [page 63]
When a level is deleted [page 63]
When a level is moved [page 64]
When a level is new [page 65]

8.10.1 When a level is unchanged

The following table shows what happens to the universe level in the different possible situations when the level is unchanged:

Table 22:

When the universe level	What happens to the universe level
Is unchanged	No change
Is updated	No change
Is deleted	No change. Create if the option *Regenerate Objects manually deleted* = *Yes*
Is moved	No change
Is hidden	No change

Related Information

To refresh an OLAP universe [page 47]
When the name or description of a level is updated [page 63]
When a level is deleted [page 63]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

When a level is moved [page 64]
When a level is new [page 65]

8.10.2 When the name or description of a level is updated

The following table shows what happens to the universe level in the different possible situations when the name or description of a level is updated:

Table 23:

When the universe level	What happens to the universe level
Is unchanged	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.
Is updated	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.
Is deleted	No change. Create if the option: *Regenerate Objects manually deleted* = *Yes*
Is moved	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.
Is hidden	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.

Related Information

To refresh an OLAP universe [page 47]
When a level is unchanged [page 62]
When a level is deleted [page 63]
When a level is moved [page 64]
When a level is new [page 65]

8.10.3 When a level is deleted

The following table shows what happens to the universe level in the different possible situations when the level is deleted:

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Table 24:

When the universe level	What happens to the universe level
Is unchanged	Delete if the option *Delete obsolete objects* is selected. Make the sub-class obsolete if the option *Hide obsolete objects* is selected.
Is updated	Delete if the option *Delete obsolete objects* is selected. Make the sub-class obsolete if the option *Hide obsolete objects* is selected.
Is deleted	No change.
Is moved	Delete if the option *Delete obsolete objects* is selected. Make the sub-class obsolete if the option *Hide obsolete objects* is selected.
Is hidden	Delete if the option *Delete obsolete objects* is selected. Make the sub-class obsolete if the option *Hide obsolete objects* is selected.

Related Information

To refresh an OLAP universe [page 47]
When a level is unchanged [page 62]
When the name or description of a level is updated [page 63]
When a level is moved [page 64]
When a level is new [page 65]

8.10.4 When a level is moved

The following table shows what happens to the universe level in the different possible situations when the level is moved:

Table 25:

When the universe level	What happens to the universe level
Is unchanged	Move the level accordingly (within the same hierarchy).
Is updated	Move the level accordingly (within the same hierarchy).
Is deleted	No change. Create if the option *Regenerate Objects manually deleted* = *Yes*.
Is moved	No change. Create if the option *Regenerate Objects manually deleted* = *Yes*.
Is hidden	Move the level accordingly (within the same hierarchy).

Related Information

To refresh an OLAP universe [page 47]
When a level is unchanged [page 62]
When the name or description of a level is updated [page 63]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

When a level is deleted [page 63]
When a level is new [page 65]

8.10.5 When a level is new

The universe level is created when the OLAP level is created.

Related Information

To refresh an OLAP universe [page 47]
When a level is unchanged [page 62]
When the name or description of a level is updated [page 63]
When a level is deleted [page 63]
When a level is moved [page 64]

8.11 How SAP variables are managed in OLAP universe updates

This section only concerns SAP data sources. The universe filter and associated list of values objects are the equivalent of the OLAP variable. How the universe objects are managed with respect to the OLAP objects depends on the type of change. Refer to the topics listed below to see how the universe objects are impacted by specific OLAP object changes.

Related Information

To refresh an OLAP universe [page 47]
When an SAP variable is unchanged [page 65]
When the name or description of an SAP variable is updated [page 66]
When an SAP variable is deleted [page 67]
When an SAP variable is new [page 68]

8.11.1 When an SAP variable is unchanged

The following table shows how universe filters are managed in the different possible situations when the SAP source variable is unchanged:

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Table 26:

When the universe filter	What happens to the universe filter
Is unchanged	No change
Is updated	No change
Is deleted	Create. If the characteristic referenced in the variable is not in the universe, then create a sub-class for that characteristic too.
Is moved	No change
Is hidden	No change

Related Information

To refresh an OLAP universe [page 47]
When the name or description of an SAP variable is updated [page 66]
When an SAP variable is deleted [page 67]
When an SAP variable is new [page 68]

8.11.2 When the name or description of an SAP variable is updated

The following table shows how universe filters are managed in the different possible situations when either the name or description of the SAP source variable is updated:

Table 27:

When the universe filter	What happens to the universe filter
Is unchanged	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.
Is updated	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.
Is deleted	Create. If a characteristic referenced in the variable is not in the universe, then create a sub-class for that characteristic too.
Is moved	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

When the universe filter	What happens to the universe filter
Is hidden	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Remain unchanged if these options are selected.

Related Information

To refresh an OLAP universe [page 47]
When an SAP variable is unchanged [page 65]
When an SAP variable is deleted [page 67]
When an SAP variable is new [page 68]

8.11.3 When an SAP variable is deleted

The following table shows how universe filter is managed in the different possible situations when the SAP variable is deleted:

Table 28:

When the universe filter	What happens to the universe filter
Is unchanged	Delete if the option *Delete obsolete objects* is selected. Make the sub-class hidden if the option *Hide obsolete objects* is selected. When made obsolete: also change from Compulsory to Optional to avoid automatic application in queries
Is updated	Delete if the option *Delete obsolete objects* is selected. Make the sub-class hidden if the option *Hide obsolete objects* is selected. When made obsolete: also change from Compulsory to Optional to avoid automatic application in queries
Is deleted	No change.
Is moved	Delete if the option *Delete obsolete objects* is selected. Make the sub-class hidden if the option *Hide obsolete objects* is selected. When made obsolete: also change from Compulsory to Optional to avoid automatic application in queries
Is hidden	Delete if the option *Delete obsolete objects* is selected. Make the sub-class hidden if the option *Hide obsolete objects* is selected. When made obsolete: also change from Compulsory to Optional to avoid automatic application in queries

Related Information

To refresh an OLAP universe [page 47]
When an SAP variable is unchanged [page 65]
When the name or description of an SAP variable is updated [page 66]
When an SAP variable is new [page 68]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

8.11.4 When an SAP variable is new

The following table shows how universe filters are managed in the different possible situations when the SAP variable is new:

Table 29:

When the universe filter	What happens to the universe filter
Is unchanged	Create. If a characteristic referenced in the variable is not in the universe, then create a sub-class for that characteristic too.
Is updated	Create. If a characteristic referenced in the variable is not in the universe, then create a sub-class for that characteristic too.
Is deleted	Create. If a characteristic referenced in the variable is not in the universe, then create a sub-class for that characteristic too.
Is moved	Create. If a characteristic referenced in the variable is not in the universe, then create a sub-class for that characteristic too.
Is hidden	Create. If a characteristic referenced in the variable is not in the universe, then create a sub-class for that characteristic too.

Related Information

To refresh an OLAP universe [page 47]
When an SAP variable is unchanged [page 65]
When the name or description of an SAP variable is updated [page 66]
When an SAP variable is deleted [page 67]

8.12 How key figures or measures are managed in OLAP universe updates

SAP data souces use key figures, MSAS and Essbase data sources use measures. The universe measure is the equivalent of the OLAP key figure. How the universe objects are managed with respect to the OLAP objects depends on the type of change. Refer to the topics listed below to see how the universe objects are impacted by specific OLAP object changes.

Related Information

To refresh an OLAP universe [page 47]
When a key figure or measure is unchanged [page 69]
When the name, description, or data type of a key figure or measure is updated [page 69]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

When a key figure or measure is deleted [page 70]
When a key figure or measure is moved [page 71]
When a key figure or measure is new [page 72]

8.12.1 When a key figure or measure is unchanged

The following table shows what happens to the universe measure in the different possible situations when the SAP key figure or MSAS/Essbase measure is unchanged:

Table 30:

When the universe measure	What happens to the universe measure
Is unchanged	No change
Is updated	No change
Is deleted	No change. Create if the option *Regenerate Objects manually deleted* = *Yes*
Is moved	No change
Is hidden	No change

Related Information

To refresh an OLAP universe [page 47]
When the name, description, or data type of a key figure or measure is updated [page 69]
When a key figure or measure is deleted [page 70]
When a key figure or measure is moved [page 71]
When a key figure or measure is new [page 72]

8.12.2 When the name, description, or data type of a key figure or measure is updated

The following table shows what happens to the universe measure in the different possible situations when the SAP key figure or MSAS/Essbase measure is updated:

Table 31:

When the universe measure	What happens to the universe measure
Is unchanged	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Update the data type if the option *Keep data type of objects* is not selected. Remain unchanged if these options are selected.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

When the universe measure	What happens to the universe measure
Is updated	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Update the data type if the option *Keep data type of objects* is not selected. Remain unchanged if these options are selected.
Is deleted	No change. Create if the option *Regenerate Objects manually deleted* = *Yes*
Is moved	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Update the data type if the option *Keep data type of objects* is not selected. Remain unchanged if these options are selected.
Is hidden	Update the business name if the option *Keep business name* is not selected. Update the description if the option *Keep description* is not selected. Update the data type if the option *Keep data type of objects* is not selected. Remain unchanged if these options are selected.

Related Information

To refresh an OLAP universe [page 47]
When a key figure or measure is unchanged [page 69]
When a key figure or measure is deleted [page 70]
When a key figure or measure is moved [page 71]
When a key figure or measure is new [page 72]

8.12.3 When a key figure or measure is deleted

The following table shows what happens to the universe measure in the different possible situations when the SAP key figure or MSAS/Essbase measure is deleted:

Table 32:

When the universe measure	What happens to the universe measure
Is unchanged	Delete if the option *Delete obsolete objects* is selected. Make the sub-class obsolete if the option *Hide obsolete objects* is selected.
Is updated	Delete if the option *Delete obsolete objects* is selected. Make the sub-class obsolete if the option *Hide obsolete objects* is selected.
Is deleted	No change.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

When the universe measure	What happens to the universe measure
Is moved	Delete if the option *Delete obsolete objects* is selected. Make the sub-class obsolete if the option *Hide obsolete objects* is selected.
Is hidden	Delete if the option *Delete obsolete objects* is selected. Make the sub-class obsolete if the option *Hide obsolete objects* is selected.

Related Information

To refresh an OLAP universe [page 47]
When a key figure or measure is unchanged [page 69]
When the name, description, or data type of a key figure or measure is updated [page 69]
When a key figure or measure is moved [page 71]
When a key figure or measure is new [page 72]

8.12.4 When a key figure or measure is moved

The following table shows what happens to the universe measure in the different possible situations when the SAP key figure or MSAS/Essbase measure is moved:

Table 33:

When the universe measure	What happens to the universe measure
Is unchanged	Move the object accordingly.
Is updated	Move the object accordingly.
Is deleted	No change. Create if Option: *Regenerate Objects manually deleted* = *Yes*.
Is moved	No change.
Is hidden	Move the object accordingly.

Related Information

To refresh an OLAP universe [page 47]
When a key figure or measure is unchanged [page 69]
When the name, description, or data type of a key figure or measure is updated [page 69]
When a key figure or measure is deleted [page 70]
When a key figure or measure is new [page 72]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

8.12.5 When a key figure or measure is new

The universe measure is created when the OLAP key figure or measure is created.

Related Information

To refresh an OLAP universe [page 47]
When a key figure or measure is unchanged [page 69]
When the name, description, or data type of a key figure or measure is updated [page 69]
When a key figure or measure is deleted [page 70]
When a key figure or measure is moved [page 71]

8.13 How SAP key dates are managed in OLAP universe updates

This section applies to SAP data sources only. The universe parameter is the equivalent of the OLAP Key Date. How the universe objects are managed with respect to the OLAP objects depends on the type of change. Refer to the topics listed below to see how the universe objects are impacted by specific OLAP object changes.

Related Information

To refresh an OLAP universe [page 47]
When a SAP key date is unchanged [page 72]
When a SAP key date is deleted [page 73]
When a SAP key date is new [page 73]

8.13.1 When a SAP key date is unchanged

The universe parameter is the equivalent of the OLAP Key Date. The following table shows what happens to the universe parameter in the different possible situations when the SAP key date is unchanged:

Table 34:

When the universe parameter	What happens to the universe parameter
Is unchanged	No change
Is updated	Not applicable

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

When the universe parameter	What happens to the universe parameter
Is deleted	Not applicable
Is moved	Not applicable
Is hidden	Not applicable

Related Information

To refresh an OLAP universe [page 47]
When a SAP key date is deleted [page 73]
When a SAP key date is new [page 73]

8.13.2 When a SAP key date is deleted

The universe parameter is the equivalent of the OLAP Key Date. The following table shows what happens to the universe parameter in the different possible situations when the SAP key date is deleted:

Table 35:

When the universe parameter	What happens to the universe parameter
Is unchanged	Delete
Is updated	Not applicable
Is deleted	Not applicable
Is moved	Not applicable
Is hidden	Not applicable

Related Information

To refresh an OLAP universe [page 47]
When a SAP key date is unchanged [page 72]
When a SAP key date is new [page 73]

8.13.3 When a SAP key date is new

The universe parameter is the equivalent of the OLAP Key Date. The following table shows what happens to the universe parameter in the different possible situations when the SAP key date is new:

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Table 36:

When the universe parameter	What happens to the universe parameter
Is unchanged	Create
Is updated	Not applicable
Is deleted	Create
Is moved	Not applicable
Is hidden	Not applicable

Related Information

To refresh an OLAP universe [page 47]
When a SAP key date is unchanged [page 72]
When a SAP key date is deleted [page 73]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

9 How the different OLAP cubes are mapped to universes

9.1 How SAP BW objects are mapped and used in a universe

When you create a universe from either an InfoCube or a BEx Query, the universe design tool maps SAP BW OLAP structures to equivalent classes and objects in the universe.

All InfoObjects in the BEx Query set as rows, columns, free characteristics, and filters are exposed to the universe. This includes characteristics, hierarchies, key figures, structures, and variables.

Hierarchies are mapped, allowing Web Intelligence users to drill down according to BW hierarchies.

For InfoCubes, all the dimensions, key figures, and hierarchies are mapped.

The following table shows the universe objects created for each BW object.

Table 37:

SAP BW object:	Universe objects created:
Dimension Group	Class
Characteristic	Subclass with dimension and detail objects
Characteristic with hierarchy	If data source is a BEx Query: Subclass containing dimension and detail objects for each hierarchy level in the currently defined hierarchy If data source is an InfoCube: Subclasses containing dimension and detail objects for each hierarchy level for all hierarchies defined for the characteristic
Structure based on Characteristics (BEx Queries only)	Class with single dimension object for the structure
Navigation attribute	Subclass with dimension and detail objects (same as Characteristic)
Display attribute	Detail object for the dimension
Key Figure structure	Class
Key Figure	Measure object in the class for the Key Figure structure with dimension objects for units/currency.
Calculated Key Figure (BEx Queries only)	Measure and dimension objects (same as Key Figure)
Restricted Key Figure (BEx Queries only)	Measure and dimension objects (same as Key Figure)

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

SAP BW object:	Universe objects created:
Variables (BEx Queries only)	Filter mandatory in query In the class for the dimension to which the variable applies, two dimension objects supporting the list of values, one for caption, one for description.
Key date variable (BEx Queries only)	Universe parameters defining key date variable in the universe

Characteristics in the Filters section of the BEx Query are not mapped. However, the filtering applies to the universe. If the filter has a fixed value, the filter is applied transparently when running the Web Intelligence query. If the characteristic has a variable defined, the variable is mapped with these limitations:

- The variable always behaves like a mandatory variable
- Hierarchy and hierarchy node variables are supported except for the hierarchy version variable

To avoid these limitations, move the characteristic from the Filter section to the Free section of the BEx Query.

Related Information

How characteristics are mapped and used [page 76]
How key figures are mapped and used [page 77]
How hierarchies are mapped and used [page 77]
How variables are supported in universes [page 78]
How variables are mapped to a universe [page 80]

9.1.1 How characteristics are mapped and used

When no hierarchy is defined on the characteristic in the BEx Query or InfoCube, the universe design tool creates a class containing the characteristic as two dimension objects: Level 00 and Level 01. The Level 00 dimension represents the aggregation of the characteristic when all members are selected (the member returned from SAP NetWeaver technology platform is *All members*). The Level 01 dimension contains all members for the characteristic as a flat list of values.

For each dimension object, the universe design tool creates a detail object for the key, up to three detail objects for the description (short, medium, and long descriptions), and a detail object for each display attribute.

The SELECT clause is defined using the technical name of the characteristic.

Navigation attributes defined in the BW Query are mapped in the parent object class in the same way as characteristics are mapped.

> **Note**
>
> A large number of navigation attributes defined in the universe negatively impacts the performance of the query in Web Intelligence.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Stuctures defined in the BEx Query that are based on characteristics are included in the universe as single-dimension objects with the elements of the structure as dimension members.

9.1.2 How key figures are mapped and used

All key figures in the InfoCube or defined in the BEx Query are included in the universe under a single object class called Key Figures.

Most key figures are defined in BW with either a currency or a unit characteristic. For each key figure, the universe design tool creates:

- A measure object with numeric format corresponding to the key figure without the unit.
- A dimension object with character format that contains the unit or currency. For example, 'USD', '€', 'km'.
- A dimension object with character format that contains the key figure and the unit (formatted value) based on user preferences configured on the SAP server. For example, '200 USD', '345 €', '25 km'.

The Key Figures class includes the calculated key figures and restricted key figures defined in the BEx Query. The original calculation and restrictions are applied to the query, but are not exposed in the universe.

9.1.3 How hierarchies are mapped and used

Hierarchies are mapped to allow Web Intelligence users to drill down with SAP BW hierarchies in the same way as custom-made universe hierarchies.

> **Note**
>
> The *Use Query Drill* option in the Web Intelligence Document Properties dialog box significantly improves drill down performance.

When a hierarchy is defined on a characteristic in the BEx Query, the universe design tool creates one hierarchical structure in the universe, with a subclass for each level in the hierarchy. The structure depends on the current BEx Query definition:

- If a hierarchy is defined in the BEx Query, the universe design tool creates this hierarchy structure in the universe.
- If a hierarchy variable is defined in the BEx Query that allows the user to choose a hierarchy at run time, the universe design tool creates a generic hierarchy in the universe. The structure has the highest number of levels defined for any of the hierarchy structures available for the characteristic.

When building a universe on top of an InfoCube, all hierarchies defined on the characteristic are exposed in the resulting universe. The universe design tool creates subclasses for each hierarchical structure, each containing subclasses for the levels in that hierarchy.

In the universe, Level 00 of a hierarchy represents the top node of the structure. When multiple tops exist for the hierarchical structure, the Level 00 dimension contains all top nodes as a list of values. When the hierarchy attribute is set to not filter unassigned nodes, it is necessary to include Level 00 with the top node for unassigned members. Unassigned members are grouped at the lowest level of the hierarchy.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

> **Note**
>
> Most often SAP BW hierarchies have only one top node. You can delete the Level 00 objects from the default universe to make the universe simpler to use. Generally, you only need to keep Level 00 when there is a need to query/report unassigned members.

If the number of levels in a hierarchy changes in the BEx Query, you must update the universe.

Related Information

About OLAP universe lifecycle management [page 44]

9.1.4 How variables are supported in universes

SAP variables can be interpreted as user prompts defined in the BW Query. Variables can be mandatory or optional, and can have default values.

Variables for characteristics are used to filter values for a characteristic. Variables are populated with values when a query is executed. They can store characteristic values, hierarchies, hierarchy nodes, texts, and formula elements.

SAP BW variables apply to BEx Queries only.

> **Note**
>
> When defining the variable in the Query Designer, on the SAP BW Variables Wizard Other Settings dialog box, the Ready for Input option must be selected.

The following types of SAP BW variables are supported in universes:

- Characteristic variables
- Hierarchy variables, except for the hierarchy version variable
- Hierarchy node variables
- Currency variables
- Formula variables
- Text variables (as replacement path)
- Key date variables

The following table shows universe support for user entry BW variables. User entry variables can be mandatory or optional, and can have default values.

Table 38:

Variable Type		Support Level
Characteristic (including key date and currency)	single value prompt	supported
	multiple single value prompt	supported

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Variable Type		Support Level
	interval prompt	supported this is not supported for the key date variable which is a single value variable
	selection option prompt	supported as interval prompt this is not supported as an interval prompt for the key date variable which is a single value variable
	pre-calculated value set	not supported
Text		supported
Formula		price, quota, and numeric values supported
Hierarchy		supported except for version variable
Hierarchy node		supported

The following table shows universe support for other processing types of BW variables.

Table 39:

Variable type	Processing Type			
	Replacement path	Authorization	Customer exit	SAP exit
Characteristic	supported	supported	supported, no prompt is created in the universe	supported
Text	supported	N/A	supported	N/A
Formula	supported	N/A	supported	supported without user entry
Hierarchy	N/A	N/A	supported	supported
Hierarchy node	N/A	N/A	supported	supported without user entry

The Exclude operator is supported, however Web Intelligence does not specify that the selected value be excluded from the query. Other operators, such as Less than and Greater than, can only be used with Selection option entry type. The selection option type is turned into an interval for Web Intelligence prompting.

> **Note**
>
> To process BW variables in Web Intelligence, you must include at least one measure in the Web Intelligence query.

Related Information

How variables are mapped to a universe [page 80]
How key date variables are supported in a universe [page 81]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

How hierarchy and hierarchy node variables are supported in a universe [page 82]

9.1.4.1 How variables are mapped to a universe

The user needs to be prompted for all optional and mandatory variables, even when the dimension is not used in the result set, allowing the user to limit the result set. Therefore, a variable defined in the BEx Query is mapped, even if the corresponding characteristic is not in the query.

The user needs to know if a variable is mandatory or optional, and be able to ignore optional variables. Optional variables are defined as optional in the universe, and become optional prompts in Web Intelligence. Mandatory variables become mandatory prompts in Web Intelligence.

For characteristic variables, the universe design tool creates a mandatory filter in the universe. A mandatory filter is a pre-defined query filter object which is hidden to Web Intelligence users, but is applied systematically and transparently to all Web Intelligence queries built on the universe.

Table 40:

Variable Type	Mapped to
Characteristic variable, including currency and formula variable	Universe mandatory filter
Hierarchy variable	Universe mandatory filter
Hierarchy node variable	Class mandatory filter
Key date variable	Universe parameters

For each mandatory filter, two dimension objects are created as reference objects for the @Prompt function to display the expected list of values. The list of values dimensions are hidden in the universe. They are necessary for the correct functioning of the prompt so must not be deleted and must be moved or modified carefully.

Default values for variables are defined in the @Prompt function in the filter using the primary key, persistent/not persistent, and default values parameters. The @Prompt function syntax can be seen in the Properties page of the filter in the universe.

To avoid conflict between BW variables and filters defined by Web Intelligence users, objects involved in an SAP variable definition are generated with the option *Can be used in Condition* unchecked in the *Advanced* page of the object properties. This restricts Web Intelligence users from including dimensions involved in SAP variables in the Filter pane.

Example

WHERE clause generated for an SAP BW variable

This example shows the WHERE clause generated for a BW variable on dimension object Customer2. The syntax for the generated WHERE clause for a variable can be seen on the Properites page of the filter.

```
<FILTER KEY="[Z_VAR002]">
    <CONDITION OPERATORCONDITION="Equal">
        <CONSTANT TECH_NAME="@Prompt(
            'Customer Variable Single Value Mandatory',
            'A',
            'Customer2\LovCustomer Variable Single Value MandatoryBase',
            mono,
            primary_key)"/>
```


© 2015 SAP SE or an SAP affiliate company. All rights reserved.

```
    <CONDITION>
</FILTER>
```

The prompt text is generated from the BW variable name. You can edit the text to make it more descriptive.

`Customer2\LovCustomer Variable Single Value MandatoryBase` is the name of the hidden universe object that is used to build the list of values.

> **Note**
>
> If you rename the class or move the list of values object to another folder, you must update the syntax in the filter key.

9.1.4.2 How variables and lists of values are supported

A BEx Query can contain more than ten variables, which means that ten or more lists of values can be loaded. Loading and refreshing lists of values can have an important impact on performance. The following options are available for improving query performance for queries with variables:

- At universe generation time, all SAP BW variables (except key date) are mapped to mandatory filters. By default, the filter objects are not associated with a list of values (except for hierarchy node variables). You must explicitly associate a list of values in the object properties page.
- Optional variables are generated as optional prompts. An optional prompt does not automatically load the list of values at query run time.
- The delegate search option on the list of values properties presents the user with an empty list of values at query run time. The user enters search criteria to limit the number of values returned in the list of values. To activate the delegated search option for a list of values, edit the list of values properties on the object properties page of the object to which the list of values applies.

  > **Note**
  >
  > The delegated search is not supported for cascading lists of values.

Related Information

Optional prompts in OLAP universes [page 42]

9.1.4.3 How key date variables are supported in a universe

A key date variable in a BEx Query allows you to specify a date for time-dependent data. Key dates can influence the data retrieved for a dimension, for example, a product description can change over time. A key date can influence a hierarchy structure, for example, a specific cost center can be on Level 01 in one year, and on Level 02 in a different year.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

The key date variable is a special SAP BW variable because the date value entered by the user is not contained in any dimension of the BW Query. The key date is a property of the query.

In a BEx Query, the key date variable can be defined for two uses:

- To specify the valid date for a specific hierarchy, impacting only that hierarchy.
- To specify a date for the complete query. In this case, the key date that is set in a query influences the following:
  - time-dependent master data
  - currency exchange rates
  - the list of hierarchies
  - time-dependent hierarchy structures

> **Note**
>
> In the universe, the use of a key date is limited to the whole universe. Therefore, the key date generated in a universe impacts all other SAP variables and data.

SAP BW supports only one key date variable per BW Query, therefore a universe contains only one key date variable.

Key date variables can be mandatory or optional, and can have a default value. If no default value is defined and the user does not enter a value, the query uses the current system date.

The key date variable properties of the query are mapped to five universe parameters, described in the following table.

Table 41:

Parameter	Description
KEYDATE_ENABLED	Set to *Yes* if a key date is enabled on the universe.
KEYDATE_NAME	Technical name of the key date variable.
KEYDATE_CAPTION	Caption for the key date variable presented when prompting the user for a value.
KEYDATE_DEFAULT_VALUE	Default value for the key date, if it exists.
KEYDATE_MANDATORY	Set to *Yes* if a user must enter a value or use the default.

At query run time, Web Intelligence proposes the same key date for all queries. The user can modify the key date. A *Keydate Properties* dialog box is available to manage which key date is used. The user is prompted for the key date before any other type of variable prompt.

9.1.4.4 How hierarchy and hierarchy node variables are supported in a universe

A hierarchy variable is used to prompt the user for the hierarchy to be used in the query. Web Intelligence users can create queries and reports to retrieve and display members from any hierarchy.

If the hierarchy variable is optional and the user leaves the prompt empty, no hierarchy is used in the report.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

A report contains the largest number of hierarchy levels independent of the hierarchy that is selected. Hierarchy levels that are not returned in the result set are empty in the report.

A hierarchy node variable is used to prompt the user for the node to be defined as top node for the hierarchy in the query.

When a query contains both a hierarchy and hierarchy node variable, the Web Intelligence user must first select a hierarchy in the list of available hierarchies. Next, the user selects the hierarchy node. The list of hierarchy nodes available shows hierarchy nodes for all hierarchies. The list is not filtered based on the hierarchy selected. The user is responsible for selecting a node from the correct hierarchy. Selecting a hierarchy node from a different hierarchy can cause the report to be empty.

Related Information

How hierarchies are mapped and used [page 77]

9.2 How Essbase cubes are mapped to universe components

The universe design tool creates a universe from an Essbase cube by mapping Essbase outlines to equivalent classes and objects. You identify the cube data source when you create the connection.

Essbase alias tables define a set of alternate names for dimensions, levels, and members in the outline. The universe design tool generates the universe using the names according to the alias table you select when you create the connection to the Essbase data source.

In an Essbase outline, measures are defined as dimensions. You select the dimension to use as the measures dimension when you create the connection to the Essbase data source. The universe design tool generates the members of that dimension as measures in the universe.

Any dimension supports hierarchies with multiple levels. A maximum of one hierarchy can be defined for each dimension.

The following table shows which objects are created in the universe for each Essbase outline element.

Table 42:

Essbase Object	Universe object created:
Dimension	A class containing the generations for the dimension.
Generation	An object in the dimension class with two detail objects: one for caption, one for name.
Measures dimension	A class named according to the dimension selected as the measures dimension in the universe connection (usually Measures class or Accounts class).
Measure	A measure object in the measure class or subclass. The measures are created with a structure of class and subclass that matches the structure in the Essbase outline.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Measures are generated with the aggregation projection function set to `Database delegated` by default. When refreshing the Web Intelligence report, the aggregation of the measure is delegated to the database server.

Related Information

About connections to OLAP data sources [page 20]
Database delegated projection function [page 31]

9.3 How MSAS cubes are mapped to universe components

The universe design tool creates a universe from MSAS cubes by mapping MSAS structures to equivalent classes and objects. You specify the cube data source when you create the connection.

The following table shows which objects are created in the universe structures for each MSAS object. This mapping applies to MSAS virtual cubes and local cubes (.cub files) as well as MSAS standard cubes.

Table 43:

MSAS Object:	Universe object created:
Dimension	A class containing objects for that dimension.
Display Folder (MSAS 2005)	A subclass in the dimension class.
Hierarchy	A subclass in the corresponding dimension class or a sub-sub-class in the corresponding display folder class.
Attribute (MSAS 2005)	A subclass in the corresponding dimension class or a sub-sub-class in the corresponding display folder class.
Measures	A Measure class containing all measure objects. Measure objects are created in the Measure class, or the subclass for the Measure Group.
Measure Group (MSAS 2005)	A subclass in the Measure class.
Level	An object in the dimension class or subclass, and a Level All object which represents the aggregate of all sub-levels.
Level Property	A detail in the level object to which it applies.

Measures are generated with the aggregation projection function set to `Database delegated` by default. When refreshing the Web Intelligence report, the aggregation of the measure is delegated to the database server.

Related Information

About connections to OLAP data sources [page 20]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Database delegated projection function [page 31]

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

Important Disclaimers and Legal Information

Coding Samples

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended to better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, unless damages were caused by SAP intentionally or by SAP's gross negligence.

Accessibility

The information contained in the SAP documentation represents SAP's current view of accessibility criteria as of the date of publication; it is in no way intended to be a binding guideline on how to ensure accessibility of software products. SAP in particular disclaims any liability in relation to this document. This disclaimer, however, does not apply in cases of wilful misconduct or gross negligence of SAP. Furthermore, this document does not result in any direct or indirect contractual obligations of SAP.

Gender-Neutral Language

As far as possible, SAP documentation is gender neutral. Depending on the context, the reader is addressed directly with "you", or a gender-neutral noun (such as "sales person" or "working days") is used. If when referring to members of both sexes, however, the third-person singular cannot be avoided or a gender-neutral noun does not exist, SAP reserves the right to use the masculine form of the noun and pronoun. This is to ensure that the documentation remains comprehensible.

Internet Hyperlinks

The SAP documentation may contain hyperlinks to the Internet. These hyperlinks are intended to serve as a hint about where to find related information. SAP does not warrant the availability and correctness of this related information or the ability of this information to serve a particular purpose. SAP shall not be liable for any damages caused by the use of related information unless damages have been caused by SAP's gross negligence or willful misconduct. All links are categorized for transparency (see: http://help.sap.com/disclaimer).

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

www.sap.com/contactsap

© 2015 SAP SE or an SAP affiliate company. All rights reserved.
No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company. The information contained herein may be changed without prior notice.
Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.
These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.
SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.
Please see http://www.sap.com/corporate-en/legal/copyright/index.epx for additional trademark information and notices.