



User Guide: SAP BusinessObjects Analysis, edition for Microsoft Office

■ Release 1.2

2011-10-28

Copyright

© 2011 SAP AG. All rights reserved. SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries. Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company. Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase, Inc. Sybase is an SAP company. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary. These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group 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.

2011-10-28

Contents

Chapter 1	About this guide.....7
1.1	Who should read this guide?.....7
1.2	User profiles.....7
1.3	About the documentation set.....7
Chapter 2	Getting Started.....9
2.1	What is SAP BusinessObjects Analysis, edition for Microsoft Office?9
2.2	Working with Analysis in Microsoft Office 2010.....10
2.3	Working with Analysis in Microsoft Excel 2007.....10
2.4	Working with Analysis in Microsoft PowerPoint 2007.....15
2.5	Working with Analysis in Microsoft Excel 2003.....18
Chapter 3	Creating Workbooks.....23
3.1	To insert a crosstab with data.....23
3.2	Defining style sets for crosstabs.....24
3.2.1	SAP cell styles.....24
3.2.2	To apply a style set.....26
3.2.3	To create a style set.....27
3.2.4	To share a style set.....27
3.2.5	To delete a style set.....28
3.3	Inserting other components.....28
3.3.1	To insert a dynamic chart.....28
3.3.2	To insert an info field.....29
3.3.3	To insert a filter.....29
3.4	Working with formulas.....30
3.4.1	To create a formula.....31
3.4.2	SAPGetData.....32
3.4.3	SAPGetDimensionDynamicFilter.....33
3.4.4	SAPGetDimensionEffectiveFilter.....33
3.4.5	SAPGetDimensionInfo.....34
3.4.6	SAPGetDimensionStaticFilter.....35
3.4.7	SAPGetDisplayedMeasures.....35

3.4.8	SAPGetInfoLabel.....	36
3.4.9	SAPGetMeasureFilter.....	36
3.4.10	SAPGetMember.....	37
3.4.11	SAPGetSourceInfo.....	37
3.4.12	SAPGetVariable.....	38
3.4.13	SAPGetWorkbookInfo.....	39
3.4.14	SAPListOf.....	40
3.4.15	SAPListOfDimensions.....	40
3.4.16	SAPListOfDynamicFilters.....	41
3.4.17	SAPListOfEffectiveFilters.....	42
3.4.18	SAPListOfMembers.....	42
3.4.19	SAPListOfMessages.....	43
3.4.20	SAPListOfStaticFilters.....	44
3.4.21	SAPListOfVariables.....	44
3.4.22	SAPSetFilterComponent.....	45
3.5	Converting crosstab cells to formula.....	46
3.5.1	To convert a crosstab to formula.....	47
3.6	Working with macros.....	48
3.6.1	SAPAddMessage.....	49
3.6.2	SAPCallMemberSelector.....	49
3.6.3	SAPExecuteCommand.....	50
3.6.4	SAPExecutePlanningFunction.....	53
3.6.5	SAPExecutePlanningSequence.....	53
3.6.6	SAPGetCellInfo.....	54
3.6.7	SAPGetProperty.....	55
3.6.8	SAPLogon.....	56
3.6.9	SAPMoveDimension.....	57
3.6.10	SAPSetFilter.....	58
3.6.11	SAPSetRefreshBehaviour.....	59
3.6.12	SAPSetVariable.....	60
3.6.13	Using Analysis functions.....	61
3.6.14	Hiding / Showing Analysis UI components.....	63
3.6.15	Enhancing the Analysis Ribbon.....	64
3.6.16	Using callbacks.....	65
3.6.17	Using LastError information.....	67
3.6.18	Syntax for entering values.....	69
Chapter 4	Analyzing Data.....	71
4.1	To open a workbook.....	71
4.2	Using a default workbook to open a data source.....	72
4.3	Analyzing data with the design panel.....	73

4.3.1	The Analysis tab.....	73
4.3.2	The Information tab.....	74
4.3.3	The Components tab.....	75
4.3.4	Managing components in the design panel.....	80
4.4	Prompting.....	82
4.4.1	To define prompt values.....	83
4.4.2	To select workbook properties for prompting.....	86
4.5	Filtering data	87
4.5.1	Filtering members.....	87
4.5.2	Filtering measures.....	92
4.5.3	To show/hide zeros in rows and columns.....	95
4.6	Sorting data.....	96
4.6.1	To sort values.....	97
4.6.2	To sort members.....	97
4.7	Working with hierarchies.....	98
4.7.1	To include dimensions with hierarchies in an analysis.....	99
4.7.2	To display single dimensions as hierarchy.....	100
4.8	Calculating new measures	101
4.8.1	To calculate a new measure based on available measures.....	101
4.8.2	To add a new measure based on one available measure.....	102
4.9	Defining Conditional Formatting.....	104
4.9.1	To define a conditional format.....	105
4.9.2	To edit conditional formats.....	106
4.10	Defining the display of members, measures and totals.....	106
4.10.1	To define the members display.....	107
4.10.2	Defining the measures display.....	107
4.10.3	Defining the totals display.....	110
4.11	To comment a data cell.....	112
4.12	To save a workbook.....	113
Chapter 5	Analyzing SAP HANA data.....	115
5.1	Analyzing SAP HANA data sources.....	115
Chapter 6	Planning Data.....	117
6.1	To plan data manually	117
6.2	To plan data with planning objects	119
6.3	To recalculate planning data	121
6.4	To save planning data	121
6.5	To lock cells	121
6.6	To undo changes while planning data	122

6.7	To switch between display and change mode.....	122
Chapter 7	Creating Presentations.....	125
7.1	To create a slide out of Microsoft Excel.....	125
7.2	Presentation settings.....	126
Chapter 8	Using Analysis in other SAP BusinessObjects Applications.....	127
8.1	To save an analysis view.....	127
8.2	To insert an analysis view.....	127
Chapter 9	Settings.....	129
9.1	User settings.....	129
9.2	Support settings.....	130
Chapter 10	Troubleshooting.....	133
10.1	To enable the Analysis Add-In after system crash (Microsoft Office 2007 and 2010).....	133
10.2	To enable the Analysis Add-In after system crash (Microsoft Excel 2003)	133
10.3	Solving issues regarding the creation of Microsoft PowerPoint slides.....	134
Appendix A	More Information.....	135

About this guide

1.1 Who should read this guide?

This guide is intended for users interested in building and analyzing workbooks using SAP BusinessObjects Analysis, edition for Microsoft Office.

1.2 User profiles

There are three user profiles for SAP BusinessObjects Analysis, edition for Microsoft Office:

- **Workbook Creator**
Users who create and maintain workbooks based on SAP BEx queries, query views and SAP NetWeaver BW InfoProvider.
- **Data Analyst**
Users who navigate through existing workbooks and analyze the data they contain. They can also include workbooks in a Microsoft PowerPoint presentation and continue the analysis there.
- **Administrator**
IT specialists who install, configure and administer SAP BusinessObjects Analysis, edition for Microsoft Office. They also assign security rights and authorizations to workbook creators and analyzers.

If your existing profile needs to be modified, contact your IT administrator.

1.3 About the documentation set

The documentation set for SAP BusinessObjects Analysis, edition for Microsoft Office, comprises the following guides and online help products:

Tip:

The guides and tutorials are regularly updated and enhanced. Make sure that you have the latest version by checking the SAP Help Portal and SAP Community Network on a regular basis.

Administrator Guide

The Administrator Guide contains detailed information that a user needs to install, configure and administer the edition for Microsoft Office. The guide is available on the SAP Help Portal.

User Guide

The User Guide contains the conceptual information, procedures and reference material that a user needs to create and analyze Microsoft Excel workbooks and Microsoft PowerPoint slides with the edition for Microsoft Office. The guide is available on the SAP Help Portal.

Online Help

The online help contains the same information as the User Guide. It can be called by pressing the **Help** button in the "Setting" group on the "Analysis" tab. To access context sensitive help, move the mouse cursor to a field in the ribbon and select F1. For dialogs, you can access context sensitive help by selecting F1 when the dialog is open.

What's New Guide

The What's New guide for SAP BusinessObjects Analysis, edition for Microsoft Office, provides a complete list of the new and modified features for SAP BusinessObjects Analysis since the previous release. The guide is available on the SAP Help Portal.

eLearning Tutorials

The tutorials show you how to use SAP BusinessObjects Analysis. They give you a quick introduction to different features so that you can learn the basics of working with the Add-In. They also give you a first impression of the look and feel. The tutorials are available in the SAP Community Network at [Analysis, Official Product Tutorials](#).

Note:

SAP BusinessObjects Analysis, OLAP edition, although related very closely to SAP BusinessObjects Analysis, edition for Microsoft Office, has its own documentation set, including its own user guide and online help.

Getting Started

2.1 What is SAP BusinessObjects Analysis, edition for Microsoft Office?

SAP BusinessObjects Analysis, edition for Microsoft Office, is a Microsoft Office Add-In that allows multidimensional analysis of OLAP sources in Microsoft Excel, MS Excel workbook application design, and intuitive creation of BI presentations with MS PowerPoint. The Add-In is available for the following Microsoft Office versions:

- Microsoft Office 2010 (Excel and PowerPoint)
- Microsoft Office 2007 (Excel and PowerPoint)
- Microsoft Office 2003 (Excel)

In the edition for Microsoft Office, you can use SAP BEx Queries, query views and SAP Netweaver BW InfoProvider as data sources. The data is displayed in the workbook in crosstabs. You can insert multiple crosstabs in a workbook with data from different sources and systems. If the workbook will be used by different users, it is also helpful to add info fields with information on the data source and filter status.

Using the design panel, you can analyze the data and change the view on the displayed data. You can add and remove dimensions and measures to be displayed easily with drag and drop. To avoid single refreshes after each step, you can pause the refresh to build a crosstab. After ending the pause, all changes are applied at once.

You can refine your analysis using conditional formatting, filter, prompting, calculations and display hierarchies. You can also add charts to your analysis. If you want to keep a status of your navigation, you can save it as an analysis view. Other users can then reuse your analysis.

For more sophisticated workbook design, the edition for Microsoft Office contains a dedicated set of functions in Microsoft Excel to access data and meta data of connected BW systems. There are also a number of API functions available that you can use with the Visual Basic Editor, to filter data and set values for BW variables.

You can also plan business data based on the current data in your data source. You can enter the planning data manually and you can enter planning data automatically using planning functions and planning sequences of SAP NetWeaver BW Integrated Planning.

Analysis, edition for Microsoft Office, must be installed on your local machine. You can connect directly to a SAP NetWeaver BW system or you can connect via a business intelligence platform (SAP BusinessObjects Enterprise or SAP BusinessObjects Business Intelligence platform) to include data sources. Typically, you use the business intelligence platform of SAP BusinessObjects to store and share workbooks in productive environments, but in test systems, you can also directly connect to a BW system.

Using the business intelligence platform enables you to save workbooks and presentations with their navigation state in a central management system and to reuse these analysis views in other applications such as SAP Crystal Reports or Analysis, OLAP edition.

To get a first impression of the look and feel of the Add-In, you can have a look at the Analysis eLearning tutorials. They are available in the SAP Community Network at [Analysis, Official Product Tutorials](#).

2.2 Working with Analysis in Microsoft Office 2010

In Microsoft Office 2010, Analysis is available as a separate tab in the ribbon for Microsoft Excel and for Microsoft Powerpoint. The tab for Analysis corresponds to the tab in Microsoft Office 2007.

There are three options that are available via the Microsoft Office button in Microsoft Office 2007. In Microsoft Office 2010, you can find these three options in the ribbon tab File -> Add-Ins:

- Open Data Source
- Open Workbook
- Save Workbook

Related Topics

- [Working with Analysis in Microsoft Excel 2007](#)
- [Working with Analysis in Microsoft PowerPoint 2007](#)

2.3 Working with Analysis in Microsoft Excel 2007

In Microsoft Excel 2007, Analysis is available as a separate tab in the ribbon. The ribbon is part of the Microsoft Office user interface above the main work area that presents commands and options. Starting in the 2007 Microsoft Office system, this replaces menus and toolbars.

This guides describes procedures using the ribbon. Most of the options are also available via the context menu.



The **Analysis** tab contains the following groups:

- Data Source
- Undo
- Data Analysis
- Display
- Insert Component
- Tools
- Planning
- Design Panel




- Settings

The following tables describe the groups and their options.



Data Source group

Icon	Description
	Insert Data Source Insert data from a source system into a crosstab. More information: To insert a crosstab with data
	Refresh All Refresh all data sources. More information: The Components tab







There are three options that are available via the Microsoft Office button:

-  Open Data Source
 More information: [Using a default workbook to open a data source](#)
-  Open Workbook
 More information: [To open a workbook](#)
-  Save Workbook
 More information: [To save a workbook](#)


Undo group




Icon	Description
	Undo Undo last Analysis step.
	Redo Redo last Analysis step.

Data Analysis group




Icon	Description
	<p>Prompts</p> <p>Enter values for query parameters and variables.</p> <p>More information: Prompting</p>
	<p>Filter</p> <p>Define filter criteria for data.</p> <p>More information: To filter data by measure / To filter data by member</p>
	<p>Sort</p> <p>Sort data.</p> <p>More information: Sorting data</p>
	<p>Hierarchy</p> <p>Define hierarchy options such as expansion level and parent member positions.</p> <p>More information: Working with hierarchies</p>
	<p>Calculations</p> <p>Define simple calculations (+, -, *, /) and dynamic calculations (for example, ranking and cumulation).</p> <p>More information: Calculating new measures</p>
	<p>Swap Axes</p> <p>Swap rows and columns.</p>

Display group


Icon	Description
	<p>Conditional Formatting</p> <p>Define rules for highlighting values using colors and symbols.</p> <p>More information: To define a conditional format</p>


Icon	Description
	<p>Member Display</p> <p>Configure display for members (key/text).</p> <p>More information: To define the members display</p>
	<p>Measure Display</p> <p>Define display options for measures (for example, decimal places, scaling factors and currencies).</p> <p>More information: Defining the measures display</p>
	<p>Totals</p> <p>Configure display, position and calculation of totals.</p> <p>More information: Defining the totals display</p>

Insert Component group

Icon	Description
	<p>Chart</p> <p>Insert dynamic chart.</p> <p>More information: To insert a dynamic chart</p>
	<p>Info Field</p> <p>Insert information on data sources (for example, name and last data update).</p> <p>More information: To insert an info field</p>
	<p>Filter</p> <p>Insert component for simple data filtering.</p> <p>More information: To insert a filter</p>







Tools group

Icon	Description
	<p>Convert to Formula</p> <p>Convert a crosstab into Excel formulas to retrieve the data.</p> <p>More information: Converting crosstab cells to formula</p>



Icon	Description
	<p>Create Slide</p> <p>Create Microsoft PowerPoint slide with data from selected crosstab.</p> <p>More information: To create a slide out of Microsoft Excel</p>

Planning group




This group is optional. You can configure in the user settings if this group should be displayed in the ribbon or not.

Icon	Description
	<p>Save</p> <p>Save plan values to InfoProvider.</p> <p>More information: To save planning data</p>
	<p>Recalculate</p> <p>Recalculate plan values.</p> <p>More information: To recalculate planning data</p>
	<p>Lock Cells</p> <p>Lock and unlock input-ready cells.</p> <p>More information: To lock cells</p>
	<p>Display</p> <p>Switch all data sources to display mode.</p> <p>More information: To switch between display and change mode</p>
	<p>Change</p> <p>Switch all data sources to change mode.</p> <p>More information: To switch between display and change mode</p>
	<p>Back</p> <p>Reset edited cells.</p> <p>More information: To undo changes while planning data</p>

Design Panel group

Icon	Description
	Display Show/hide Design Panel More information: Analyzing data with the design panel
	Pause Refresh Activate/deactivate automatic refresh after each navigation step in the Design Panel. More information: Analyzing data with the design panel

Settings group

Icon	Description
	Settings Edit settings. More information: Settings
	Style Manage crosstab styles. More information: Defining style sets for crosstabs
	Help Launch help.

2.4 Working with Analysis in Microsoft PowerPoint 2007

In Microsoft PowerPoint 2007, Analysis is available as a separate tab in the ribbon. The ribbon is part of the Microsoft Office user interface above the main work area that presents commands and options. Starting in the 2007 Microsoft Office system, this replaces menus and toolbars.



The **Analysis** tab contains the following groups :

- Data Source
- Undo
- Filter and Sort

- Display
- Insert Component
- Settings



The following tables describe the groups and their options.

Data Source group



Icon	Description
	Insert Data Source Insert data from a source system into a crosstab. More information: To insert a crosstab with data
	Refresh All Refresh all data sources. More information: The Components tab



To open and save existing presentations saved on the business intelligence platform, use the corresponding options in the Microsoft Office button.

Undo group




Icon	Description
	Undo Undo last Analysis step.
	Redo Redo last Analysis step.

Filter and Sort group


Icon	Description
	Prompts Enter values for query parameters and variables. More information: Prompting
	Filter Define filter criteria for data. More information: To filter data by measure To filter data by member


Icon	Description
	Sort Sort data. More information: Sorting data
	Hierarchy Define hierarchy options such as expansion level and parent member positions. More information: Working with hierarchies

Display group



Icon	Description
	Member Display Configure display for members (key/text). More information: To define the members display
	Measure Display Define display options for measures (for example, decimal places, scaling factors and display currency). More information: Defining the measures display
	Totals Configure display, position and calculation of totals. More information: Defining the totals display

Insert Component group



Icon	Description
	Chart Insert dynamic chart. More information: To insert a dynamic chart

Icon	Description
	<p>Info Field</p> <p>Insert information on data sources (for example, name and last data update).</p> <p>More information: To insert an info field</p>

Tools group

Icon	Description
	<p>Fit Table</p> <p>Abbreviate a table to fit one slide, or split the table across multiple slides.</p>
	<p>Move to</p> <p>Move the selected Analysis object (table, chart or info field) from its current location to different slide in the presentation.</p>

Settings group

Icon	Description
	<p>Settings</p> <p>Edit settings.</p> <p>More information: Settings</p>
	<p>Help</p> <p>Launch help.</p>

2.5 Working with Analysis in Microsoft Excel 2003










In Microsoft Excel 2003, Analysis is available as separate item in the menu. You can access all options with the menu. You can also include three toolbars: Analysis Standard, Analysis Extended and Analysis Planning. These toolbars contain most of the available options.






To toggle between showing and hiding a toolbar, choose **View > Toolbars** and select the toolbar name. A checkmark beside a toolbar name indicates that the toolbar is currently showing.

This guides describes procedures using the toolbars. Most of the options are also available via the context menu.

Analysis Standard toolbar



The Analysis Standard toolbar contains the following options:







Icon	Description
	<p>Open Workbook</p> <p>Open workbook from the business intelligence platform.</p> <p>More information: To open a workbook</p>
	<p>Open Data Source</p> <p>More information: Using a default workbook to open a data source</p>
	<p>Save Workbook</p> <p>Save workbook to the business intelligence platform .</p> <p>More information: To save a workbook</p>
	<p>Insert Data Source</p> <p>Insert data from a source system into a crosstab.</p> <p>More information: To insert a crosstab with data</p>
	<p>Refresh All</p> <p>Refresh all data sources.</p> <p>More information: The Components tab</p>
	<p>Undo</p> <p>Undo last Analysis step.</p>
	<p>Redo</p> <p>Redo last Analysis step.</p>
	<p>Prompts</p> <p>Enter values for query parameters and variables.</p> <p>More information: Prompting</p>
	<p>Charts</p> <p>Insert dynamic chart.</p> <p>More information: To insert a dynamic chart</p>

Icon	Description
	<p>Info Field</p> <p>Insert information on data sources (for example, name and last data update).</p> <p>More information: To insert an info field</p>
	<p>Filter</p> <p>Insert component for simple data filtering.</p> <p>More information: To insert a filter</p>
	<p>Convert to Formula</p> <p>Convert a crosstab into Excel formulas to retrieve the data.</p> <p>More information: Converting crosstab cells to formula</p>
	<p>Display</p> <p>Show/hide Design Panel</p> <p>More information: Analyzing data with the design panel</p>
	<p>Pause Refresh</p> <p>Activate/deactivate automatic refresh after each navigation step in the Design Panel.</p> <p>More information: Analyzing data with the design panel</p>


Analysis Extended toolbar






The Analysis Extended toolbar contains the following options:

Icon	Description
	<p>Filter</p> <p>Define filter criteria for data.</p> <p>More information: To filter data by measureTo filter data by member</p>
	<p>Sort</p> <p>Sort data.</p> <p>More information: Sorting data</p>

Icon	Description
	<p>Hierarchy</p> <p>Define hierarchy options such as expansion level and parent member positions.</p> <p>More information: Working with hierarchies</p>
	<p>Conditional Formatting</p> <p>Define rules for highlighting values using colors and symbols.</p> <p>More information: To define a conditional format</p>
	<p>Calculations</p> <p>Define simple calculations (+, -, *, /) and dynamic calculations (for example, ranking and cumulation).</p> <p>More information: Calculating new measures</p>
	<p>Member Display</p> <p>Configure display for members (key/text).</p> <p>More information: To define the members display</p>
	<p>Measure Display</p> <p>Define display options for measures (for example, decimal places, scaling factors and currencies).</p> <p>More information: Defining the measures display</p>
	<p>Totals</p> <p>Configure display, position and calculation of totals.</p> <p>More information: Defining the totals display</p>

Analysis Planning toolbar

Icon	Description
	<p>Save</p> <p>Save plan values to InfoProvider.</p> <p>More information: To save planning data</p>

Icon	Description
	Recalculate Recalculate plan values. More information: To recalculate planning data
	Lock Cells Lock and unlock input-ready cells. More information: To lock cells
	Display Switch all data sources to display mode. More information: To switch between display and change mode
	Change Switch all data sources to change mode. More information: To switch between display and change mode
	Back Reset edited cells. More information: To undo changes while planning data

Analysis menu

The Analysis menu contains all options that are available as icons in the toolbars plus the following options:

- Styles
- Settings

More information: [Settings](#)

- Help

Creating Workbooks

3.1 To insert a crosstab with data

To add a crosstab with data to a workbook, you select a data source in a SAP NetWeaver BW system. You need the appropriate authorizations for a business intelligence platform of SAP BusinessObjects and the relevant SAP NetWeaver BW systems to insert a data source in a workbook. For more information, contact your IT administrator.

You can insert SAP BEx Queries, query views and SAP Netweaver BW InfoProvider as data sources. These data sources are stored in a SAP NetWeaver BW system. You can add multiple crosstabs to worksheet or workbook. The crosstabs can contain data from the same data source or from different sources. You can also use data sources that are stored in different systems in one workbook.

1. Select the cell in the worksheet where the crosstab with the data from the selected data source should be inserted.
2. Select **Insert Data Source**.

The "Log on to SAP BusinessObjects BI platform" dialog box appears.

3. Enter your **User**, **Password** and the **WEB Service URL** to log on to the business intelligence platform.

By selecting **Skip** you can log on to a BW system directly without using the business intelligence platform. Continue with step 8 if you use this log on.

Note:

If you log on with an initial password to a BW system or your password is expired and needs to be reset, the dialog for changing the password opens automatically.

4. Optional step: Enter **System** and **Authentication**.

You will normally not be asked to supply this information. However, if you are asked to log on to a special Central Management System (CMS), you can add these two additional fields to the dialog box by selecting **Options**. Enter the name of your Central Management System in the **System** field and the authentication type in the **Authentication** field.

5. Press **OK**.

The "Select Data Source" dialog box appears.

6. Select a connection in the **Show Connections** list:

- If you select **All**, all available systems, Cubes / InfoProvider and Query / Query views on the business intelligence platform are displayed.
- If you select **System**, all available systems on the business intelligence platform are displayed.

- If you select **Cube / InfoProvider**, all available Cubes and InfoProvider on the business intelligence platform are displayed.
 - If you select **Query / Query View**, all available Queries and query views on the business intelligence platform are displayed.
 - If you select **Local System**, all systems in your local "SAP Logon" are displayed.
7. Select a system and **Next**.
To select a Query, query view or InfoProvider directly, double-click the object you want to select. The "Logon to System" dialog box appears.
 8. Enter **Client, User, Password** and **Language** in the fields and press **OK**.
 9. Select a data source in the **Select Data Source** box and press **OK**.

On the **Folders** tab, you can navigate in the **Roles** or **InfoAreas** views to find a data source.

On the **Search** tab, you can search for the description or technical name of a data source. To retrieve data sources that begin with a specific string, you can type * after a partial string.

A new crosstab with the data of the selected data source is inserted in the worksheet. You can now analyze the data and change the displayed data set according to your needs. You can also add other components to your analysis, for example charts.

You can also insert a data source using the default workbook.

Related Topics

- [Using a default workbook to open a data source](#)

3.2 Defining style sets for crosstabs

A style set is a collection of Microsoft Excel cell styles that is applied by Analysis to format the cells of a crosstab. Whenever you insert a new crosstab in a workbook, the styles in the current default style set are used to format the crosstab cells. You can change the applied style set in your analysis. With Analysis, the following style sets and their cell styles are installed:

- SAP Tradeshow Plus
- SAP Blue
- SAP Black&White

By modifying the cell styles of these style sets, you can create your own style sets and share them with other users.

3.2.1 SAP cell styles

SAP standard styles

SAP standard styles are available after the installation of the Add-In. You can modify them in the **Styles** group on the **Home** tab of Microsoft Excel. They affect the formatting as described in the following table:

Style Name	Description
SAPDimensionCell	Format for dimension header cells.
SAPMemberCell	Format for member cells (non-hierarchical dimensions).
SAPHierarchyCell	Format for hierarchical member cells (even levels 0, 2, ...).
SAPHierarchyOddCell	Format for hierarchical member cells (odd levels 1, 3, ...) .
SAPMemberTotalCell	Format for member total cells.
SAPDataCell	Format for data cells.
SAPDataTotalCell	Format for data total cells.
SAPExceptionLevel1-9	Format for highlighted cells due to conditional formats (rule priorities 1-9).
SAPEmphasized	Format for highlighted data cells (as per query definition).
SAPBorder	Format for borders around a crosstab and between header/member and data cells (format for left border is taken).
SAPEditableDataCell	Format for input-ready data cells in planning change mode.
SAPEditableDataTotal-Cell	Format for input-ready total cells in planning change mode.
SAPReadOnlyDataCell	Format for read-only data cells in planning change mode.
SAPReadOnlyDataTo-talCell	Format for read-only total cells in planning change mode.
SAPLockedDataCell	Format for locked data cells in planning change mode.
SAPLockedDataTotal-Cell	Format for locked total cells in planning change mode.

SAP custom styles

The following SAP custom styles are not available after the installation of the Add-In, but you can create them in the **Styles** group on the **Home** tab of Microsoft Excel. They affect the formatting as described in the following table:

Style Name	Description
SAPMemberCellX	Format for member cells on columns (overriding SAPMemberCell).
SAPMemberTotal-CellX	Format for member total cells on columns (overriding SAPMemberTotalCell).
SAPHierarchyCellX	Format for hierarchical member cells on columns, even levels (overriding SAPHierarchyCell).
SAPHierarchyOdd-CellX	Format for hierarchical member cells on columns, odd level (overriding SAPHierarchyOddCell).
SAPHierarchy-Cell0-9	Format for hierarchical member cells on specific level (overriding SAPHierarchyCell and SAPHierarchyOddCell).
SAPHierarchyCellX0-9	Format for hierarchical member cells on specific level on columns (overriding SAPHierarchyCellX and SAPHierarchyOddCellX).

Example: SAPMemberCellX

The column headings are defined as SAPMemberCell. If you want a different format for these cells than for member cells in rows, you can duplicate the SAPMemberCell, name it SAPMemberCellX and change the format definition. If you save this as style set, the member cells in column headings are displayed in the new defined format. The member cells in rows continue to be displayed as defined in the SAPMemberCell style.

Example: Styles for Hierarchy Levels

The standard styles for hierarchies are SAPHierarchyCell and SAPHierarchyCellOdd. With these styles, the hierarchy levels have alternating styles. If you want to have three different styles for the first three hierarchy levels of an hierarchy, create the styles SAPHierarchyCell0, SAPHierarchyCell1 and SAPHierarchyCell2. The system ignores the standard styles and applies the styles for specific styles. If an hierarchy have more than three levels, the levels after level 2 are displayed with the SAPHierarchyCell2 style.

3.2.2 To apply a style set

You can apply one of the SAP style sets or any new defined style set to a workbook.

1. Choose **Styles > Apply Style Set...**

The "Apply Style Set" dialog box appears.

2. In the list box, select the style set you want to apply.

3. Select the **Set as Default** check box if the style set should be applied as default in your workbooks. The default style set is used when you open a new workbook and insert a data source.

4. Press **OK**.

The style set is applied to all crosstabs in your workbook.

3.2.3 To create a style set

Based on available cell styles, you can define a new style set. You change the cell styles according to your needs using the Microsoft Excel style functionality. You can then save the new defined styles in a style set.

1. On the **Home** tab, in the **Styles** group, choose **Cell Styles**.

Note:

In Microsoft Excel 2003, you can find the cell styles by choosing **Format > Styles** in the menu.

The available cell styles are listed.

2. Modify the existing cell styles or create new ones according to your needs.
3. On the **Analysis** tab, in the **Settings** group, choose **Styles > Save Style Set...**
The "Save Style Set" dialog box appears.
4. Enter a **Style Set Name**.
5. Select the **Set as Default** check box if the style set should be applied as default in your workbooks.
The default style set is used when you open a new workbook and insert a data source.
6. Press **OK**.

The new defined style set is created and available in the list of style sets that can be applied to a workbook.

3.2.4 To share a style set

You can share a style set with other users by exporting the style set to a local fileshare. Other users can import the style set and use it for the analysis.

1. Apply the style set that you want to export.
2. Choose **Styles > Export Style Set...**
3. Save the style set as XML format.
The XML file contains the cell styles of the three SAP style sets and your currently applied style set.
4. Choose **Styles > Import Style Set...**
5. Select a style file from the server and press **Open**.
6. Save the imported styles as new style set.

You have exported a style set to be used by other users and / or you have imported a style set to use it in your analysis.

3.2.5 To delete a style set

You can delete all user-defined style sets. The standard SAP style set that is installed with the Add-In can not be deleted.

1. Choose **Styles > Delete User Style Set**.
The "Delete User Style Set" dialog box appears.
2. In the list box, select the style set you want to delete.
3. Press **OK**.

The style set is deleted and no longer available in the list of style sets that can be applied to a workbook.

3.3 Inserting other components

In addition to crosstabs, you can add the following components to your analysis:

- **Charts** to provide a graphical presentation of the data in the crosstab.
- **Info fields** to provide metadata information.
- **Filters** to provide simplified filtering for end users.

3.3.1 To insert a dynamic chart

1. Select a cell of the crosstab that you want to visualize in a chart.
By inserting a chart with Analysis, the data of the entire crosstab is visualized in the chart. If you want to visualize only a subset of the crosstab data, you can use Microsoft Excel functionality.
2. Press the **Chart** button.
The chart is added to the analysis. You can position it in the worksheet using drag and drop.
3. Modify the chart.
To modify the chart, you can use Microsoft Excel options for charts. For example, you can change the chart type or define a data range for the chart.
4. You can move the chart to another worksheet in the workbook.
On the **Component** tab in the design panel, select the chart you want to move, and open the **Move to** dialog. Select the sheet that should contain the chart and press **OK**.

The chart is added to the analysis according to your configuration. The chart is updated automatically when you change the displayed data in the crosstab.

3.3.2 To insert an info field

You can insert information fields to provide additional information on data displayed in the workbook sheets.

1. Select an empty cell where you want to place the info field.
2. Select the info field you want to insert.
 - Choose **Info Field** and one of the listed fields: Data Source Name, Last Data Update, Key Date, Effective Filters, Variables. If you want to insert other info fields, use the second option.

The info field is added to worksheet. If you use more than one data source in your analysis, you are prompted to select a data source.

- You can also drag and drop the info fields from the **Information** tab in the design panel to a cell in the worksheet.

Select the data source on the top of the tab and drag and drop the information you want to insert as info field. For dynamic info fields (filters and variables), you have to use the first option.

The info fields are inserted with label and source information. The functions used for the formulas are SAPGetInfoLabel and SAPGetSourceInfo. The formulas are created automatically.

3.3.3 To insert a filter

You can insert a filter component to your analysis to simplify the filtering. This helps you to quickly change the view of the displayed data, for example to different periods of time.

1. Select an empty cell where you want to place the filter component.
2. Choose **Filter** and select one of the listed dimensions to insert a filter component for this dimension.

The dimension name and a filter component formula are inserted in the worksheet. The functions used for the formulas are SAPGetDimensionInfo and SAPSetFilterComponent. The formulas are created automatically.

3. Optional Step: Specify the filter component formula.

The formula that is inserted automatically, allows the user to select multiple members for filtering. It looks like this: `=SAPSetFilterComponent("DS_2"; "0CALYEAR";"ALL")`.

You can add one of the following parameters to the formula: SINGLE, MULTIPLE, LOWERBOUNDARY, UPPERBOUNDARY to specify the filtering options. If you add the parameter SINGLE, the user can only select one member for filtering. The formula looks like this: `=SAPSetFilterComponent("DS_2"; "0CALYEAR";"ALL";"SINGLE")`.

You can also insert filter components to enable a range selection. Insert two filter components for the same dimension and add to one the parameter LOWERBOUNDARY and to the other the parameter UPPERBOUNDARY. You can now filter for the lower and upper bounds of a range.

4. Optional step: Format the filter component.

You can use the formatting options of Microsoft Excel to format cells of the filter component.

5. Select the filter icon to define a filter.

All tables on the current sheet that contain this dimension, will be filtered according to the selected filter. On the **Components** tab in the design panel, you can define which tables should be affected if not all tables should be filtered accordingly.

The filter is added to the analysis according to your configuration.

Related Topics

- [SAPGetDimensionInfo](#)
- [SAPSetFilterComponent](#)

3.4 Working with formulas

In Analysis, edition for Microsoft Office, you can use the standard functions of Microsoft Excel to build formulas. The Add-in also contains an own set of functions that you can use to build formulas. You can use these functions to include data and meta data of used data sources into your analysis. For example, you can insert information fields on data source properties, display the measure filter or list the variables of a data source. With the SAPGetData function, you can also define measure values for certain member combinations.

A Microsoft Excel formula for Analysis consists of a function and references to the data source, measures and/or dimensions. You can use the text or the key of an object to use it as reference. You can also use a cell value like B10 as reference.

The formula alias of a data source is displayed and can be changed in the data source properties on the **Components** tab in the design panel. For measures, dimensions and their members text references are better to read, but if you want to create a multi-language enabled analysis or there are duplicate texts in the meta data of your data source, you should reference these objects with their keys.

You can also use these functions in VBA macros.

Analysis functions

The following functions are available in the Analysis category:

- SAPGetData
- SAPGetDimensionDynamicFilter
- SAPGetDimensionEffectiveFilter
- SAPGetDimensionInfo
- SAPGetDimensionStaticFilter

- SAPGetDisplayedMeasures
- SAPGetInfoLabel
- SAPGetMeasureFilter
- SAPGetMember
- SAPGetSourceInfo
- SAPGetVariable
- SAPGetWorkbookInfo
- SAPListOf
- SAPListOfDimensions
- SAPListOfDynamicFilters
- SAPListOfEffectiveFilters
- SAPListOfMembers
- SAPListOfMessages
- SAPListOfStaticFilters
- SAPListOfVariables
- SAPSetFilterComponent

Related Topics

- [Using Analysis functions](#)

3.4.1 To create a formula

To create a formula with Analysis functions:

1. Select the cell in which you want to enter the formula.
2. To start the formula with a function, press the **Insert Function** button on the formula bar.
The "Insert Function" dialog box appears.
3. Select **Analysis** in the **Select a category** box.
4. Select a function.
5. Press **OK**.
The "Function Arguments" dialog box appears.
6. Enter the arguments.
To enter cell references as an argument, press the **Collapse Dialog** button (which temporarily hides the dialog box), select the cells on the worksheet, and then press the **Expand Dialog** button.
7. When you complete the formula, press **OK**.

3.4.2 SAPGetData

This function returns the measure value for a specific dimension member combination.

The formula can only return values for member combinations that are part of the current navigation state of the data source. To be part of the navigation state, the member combinations must be used in rows, columns or as a background filter. If you filter a dimension, you can only return values for member combinations that the filter contains. For example, if the navigation state of the data source displays the dimension Region in rows and the measures Sales Volume in columns, you can create a formula to return a value for a particular region, but you cannot return a value for a special customer, even if customer information is available in the data source. To be able to return values for a special customer, you have to add the dimension to the navigation state, for example as a background filter.

The formula consists of at least 3 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.

- **Measure**

Enter the name of measure, for example "Incoming Orders".

- **Member combination**

There are two methods for entering the member combination:

- Enter one parameter as member combination, for example "Region=France;Product=Services". This method is used for converting to formula.
- Enter several parameters as the member combination, for example "Region";"France";"Product";"Services". This method can only be entered manually. It is recommended for member combinations that use cell references.

Example: 3 Parameters formula

Cell H20: =SAPGetData("DS_1";"Incoming Orders";"Region=France;Product=Services")

The data for the value in this cell comes from data source DS_1. The name of the measure is Incoming Orders. The member combination is France and Services. The formula in cell H20 therefore uses the data from DS_1 to calculate the incoming orders for Region France and Product Services. If you change France to Germany in the formula, the incoming orders for Germany and Services are displayed in cell H20.

Example: >3 Parameters formula with cell reference

Cell H20: =SAPGetData("DS_1";"Incoming Orders";"Region";B10;"Product";"Services")

The data for the value in this cell comes from data source DS_1. The name of the measure is Incoming Orders. The member combination is the region entered in cell B10 and Services. For example, if you enter Spain in cell B10, the formula in cell H20 uses the data from DS_1 to calculate the incoming orders for Region Spain and Product Services. If you change Spain to France in cell B10, the incoming orders for France and Services are displayed in cell H20.

3.4.3 SAPGetDimensionDynamicFilter

This function returns the dynamic filter of a dimension. Dynamic filters are defined by the user.

The formula consists of 3 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias of the data source or planning function. You can set the alias when configuring the data source or planning function on the **Components** tab in the design panel.

- **Dimension**

Enter the technical name of the dimension.

- **Member Display**

You can enter TEXT or KEY to define how the filtered members should be displayed in the workbook.

Example:

Cell F20: =SAPGetDimensionDynamicFilter("DS_1";"0DIVISION";"TEXT")

You add a filter for dimension 0DIVISION and the following members are displayed in the analysis: Paints, Lighting, Foods. If you enter the formula in cell F20, the three filtered members are displayed in cell F20 as text.

3.4.4 SAPGetDimensionEffectiveFilter

This function returns all effective filters of a dimension: Dynamic filters defined by the user, static filters defined in the underlying source, and filters by measure defined for the selected dimension.

The formula consists of 3 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias of the data source or planning function. You can set the alias when configuring the data source or planning function on the **Components** tab in the design panel.

- **Dimension**

Enter the technical name of the dimension.

- **Member Display**

You can enter TEXT or KEY to define how the filtered members should be displayed in the workbook.

Example:

Cell F20: =SAPGetDimensionEffectiveFilter("DS_1";"0DIVISION";"TEXT")

If you enter the formula in cell F20, the members of 0DIVISION that are currently filtered by the user, the static filters that are defined in the data source and the filters by measure for this dimension are displayed in cell F20 as text. If no static filters are defined for the data source, only the dynamic filter members and filters by measure are displayed.

3.4.5 SAPGetDimensionInfo

This function returns the name of a dimension or the name of an active hierarchy.

The formula consists of 3 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.

- **Dimension**

Enter the technical name of the dimension.

- **Property Name**

You can enter the following property names:

- NAME
- ACTIVEHIERARCHY

Example:

Cell F20: =SAPGetDimensionInfo("DS_1";"0DIVISION";"NAME")

If you enter the formula in cell F20, the name of dimension 0DIVISION is displayed in cell F20.

3.4.6 SAPGetDimensionStaticFilter

This function returns the static filter of a dimension. Static filters are defined in the underlying source and cannot be changed by the user.

The formula consists of 3 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.

- **Dimension**

Enter the technical name of the dimension.

- **Member Display**

You can enter TEXT or KEY to define how the filtered members should be displayed in the workbook.

Example:

Cell F20: =SAPGetDimensionStaticFilter("DS_1";"0MATERIAL";"KEY")

If you enter the formula in cell F20, the static filter of dimension 0MATERIAL is displayed in cell F20.

3.4.7 SAPGetDisplayedMeasures

This function returns a list of all measures displayed in the analysis as text.

The formula is made up of the following argument: **Data Source**.

Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.

Example:

Cell G10: =SAPGetDisplayedMeasures("DS_1")

If you enter the formula in cell G10, all measures that are currently displayed in the crosstab are listed in cell G10. If you add or remove a measure from the crosstab, the list in cell G10 is updated accordingly.

3.4.8 SAPGetInfoLabel

This function returns the language-dependant label for an info field. The property names correspond to the info fields that are available for workbook and data sources on the **Information** tab in the design panel. Using this function, the info field labels are displayed in the selected UI language. The info field values can be inserted with function SAPGetWorkbookInfo and SAPGetSourceInfo.

The formula is made up of the following argument: **Property Name**.

For workbook related info fields, you can enter the following property names:

- WorkbookName
- CreatedBy
- CreatedAt
- LastChangedAt
- LastRefreshedAt
- LogonUser

For data source related info fields, you can enter the following property names:

- DataSourceName
- LastDataUpdate
- KeyDate
- QueryTechName
- QueryCreatedBy
- QueryLastChangedBy
- QueryLastChangedAt
- InfoProviderTechName
- InfoProviderName
- System
- LogonUser

Example:

Cell D20: =SAPGetInfoLabel("System")

The label of the info field is displayed in the selected UI language, for example in English: System.

3.4.9 SAPGetMeasureFilter

This function returns a list of all filtered measures and their rules defined for a data source.

The formula is made up of the following argument: **Data Source**.

Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.

Example:

Cell G10: =SAPGetMeasureFilter("DS_1")

If you enter the formula in cell G10, all measures that have a filter definition and the corresponding rules are displayed in a list in cell G10. If you add or remove a filter to a measure, the list in cell G10 is updated accordingly.

3.4.10 SAPGetMember

This function returns the dimension member or attribute.

The formula can only return values for dimension members or attributes that are part of the current navigation state of the data source. To be part of the navigation state, the members must be used in rows, columns or as background filter. If you filter a dimension, you can only return values for members that the filter contains.

The formula consists of 3 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.

- **Dimension Member**

Enter the technical name of a dimension and assign a member key, for example "0DIVISION=R1".

- **Member Display**

You can enter TEXT or KEY to define how the filtered members should be displayed in the workbook.

Example:

Cell G15: =SAPGetMember("DS_1";"0DIVISION=R1";"TEXT")

You want to display the text for member Retail. The key for Retail is R1. If you enter the formula in cell G15, the text of member R1 (Retail) is displayed in cell G15.

3.4.11 SAPGetSourceInfo

This function returns an info field value for a data source. The info field label can be inserted using the SAPGetInfoLabel function. The property names correspond to the info field values that are available for data sources on the **Information** tab in the design panel.

The formula consists of 2 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.

- **Property Name**

You can enter the following property names:

- DataSourceName
- LastDataUpdate
- KeyDate
- QueryTechName
- QueryCreatedBy
- QueryLastChangedBy
- QueryLastChangedAt
- InfoProviderTechName
- InfoProviderName
- System
- LogonUser

Example:

Cell D20: =SAPGetInfoLabel("DataSourceName")

Cell E20: =SAPGetSourceInfo("DS_1","DataSourceName")

In cell D20, the Data Source Name label is displayed. In cell E20, the name of the data source with alias DS_1 is displayed, for example Sales Volume Europe.

3.4.12 SAPGetVariable

This function returns a the value of the specified property for a specific BW variable.

The formula consists of 3 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.

- **Variable Name**

Enter the name or technical name of the variable.

- **Property Name**

You can enter the following property names:

- VALUE
If you enter VALUE, the current value of the variable is displayed.
- VALUEASKEY
If you enter VALUEASKEY, the current value of the variable is displayed as a key.
- DESCRIPTION
If you enter DESCRIPTION, the variable name is displayed.
- ISINPUTENABLED
If you enter ISINPUTENABLED, the function checks if the variable is input-enabled.
- ISMANDATORY
If you enter ISMANDATORY, the function checks if the variable is mandatory.
- TECHNICALNAME
If you enter TECHNICALNAME, the technical name of the variable is displayed.

Example:

Cell F20: =SAPGetVariable("DS_2";"0BW_VAR";"DESCRIPTION")

If you enter the formula in cell F20, the name of variable 0BW_VAR is displayed in cell F20.

3.4.13 SAPGetWorkbookInfo

This function returns an info field value for the current workbook. The info field label can be inserted with the SAPGetInfoLabel function. The property names correspond to the info field values that are available for workbooks on the **Information** tab in the design panel.

This formula is made up of the following argument: **Property Name**.

You can enter the following property names:

- WorkbookName
- CreatedBy
- CreatedAt
- LastChangedAt
- LastRefreshedAt
- LogonUser

Example:

Cell D20: =SAPGetInfoLabel("WorkbookName")

Cell E20: =SAPGetWorkbookInfo("WorkbookName")

In cell D20, the Workbook Name label is displayed. In cell E20, the name used for saving the workbook is displayed, for example Sales in Europe.

3.4.14 SAPListOf

This function returns a list of available objects for a specific object type. The list contains two columns for every entry (Alias and Description). For object type CROSSTABS, the function returns a list with three columns (Alias, Description and data source Alias).

This formula consists of 3 parameters and is made up of the following arguments:

- **Object Type**

You can enter the following object types:

- DATASOURCES
- CROSSTABS
- PLANNINGFUNCTIONS
- PLANNINGSEQUENCES

- **Search**

You can enter a search string. Note that the search is case sensitive. If you do not enter a search string, all available objects of the selected object type will be listed.

- **Search in**

You can search in ALIAS or DESCRIPTION. This parameter is optional. The default value is ALIAS.

Example:

Cell G15: =SAPListOf("CROSSTABS";"Cross")

Starting in Cell G15, a list with three columns (crosstab Alias, crosstab Description, data source alias) is displayed with all crosstabs that contain the search string Cross in their Alias.

3.4.15 SAPListOfDimensions

This function returns a list of dimensions. Depending on the search entries, the list can display all dimensions with their current drill state, dimensions that match a defined search string with their current drill state or just the dimensions used in rows, columns or background filter.

For planning functions, the function returns a list of all dimensions defined on the aggregation level of the planning function. The axis type is always ROW.

This formula consists of 3 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias of the data source or planning function. You can set the alias when configuring the data source or planning function on the **Components** tab in the design panel.

- **Search In**

You can determine which columns the search should be executed in by selecting TECHNICALNAME, DESCRIPTION or AXIS. The default selection is TECHNICALNAME.

- **Search**

If you have selected TECHNICALNAME or DESCRIPTION, you can enter a search string. Note that the search is case sensitive. If you do not enter a search string, all available dimensions will be listed.

If you have selected AXIS, you can use the axis type to define which dimensions should be displayed:

- Enter ROWS to list the dimensions that are used in rows in the current analysis.
- Enter COLUMNS to list the dimensions that are used in columns in the current analysis.
- Enter FILTER to list the dimensions that are used as background filter in the current analysis.
- Enter ALL to list all available dimensions with their current drill state. For dimensions that are not used in the current analysis, the drill state column is empty.

If the selected axis type contains no dimensions in the current analysis, the list is empty.

Example:

Cell G11: =SAPListOfDimensions("DS_1";"TECHNICALNAME";"0SOLD")

Starting in Cell G11, a list with three columns (Technical Name, Description, Drill state) is displayed with all dimensions that contain 0SOLD in their technical name.

3.4.16 SAPListOfDynamicFilters

This function returns a list of all dynamic filters of a data source. Dynamic filters are defined by the user using *Filter by Member*.

The formula consists of 2 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.

- **Member Display**

You can enter TEXT or KEY to define how the filtered members should be displayed in the workbook.

Example:

Cell G15: =SAPListOfDynamicFilters("DS_1";"TEXT")

If you enter the formula in cell G15, the dynamic filters of data source DS_1 are displayed as text.

3.4.17 SAPListOfEffectiveFilters

This function returns a list of all effective filters for a data source.

The formula consists of 2 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.

- **Member Display**

You can enter TEXT or KEY to define how the filtered members should be displayed in the workbook.

Example:

Cell F20: =SAPListOfEffectiveFilters("DS_1";"TEXT")

You have added dimension Region from data source DS_1 to your analysis. You filter this dimension and the following members are part of the analysis: California, Arizona, Florida, Nevada. If you enter the formula in cell F20, the name of the dimension is displayed in cell F20, and the four filtered members are listed as text in cell G20.

3.4.18 SAPListOfMembers

This functions returns a list of members for a selected dimension or prompt. Each list entry contains the key, text and key as filter string of a member. The key as filter string differs from the key if the member is compounded to another dimension.

The formula consists of 5 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias of the data source or planning object. You can set the alias when configuring the data source or planning object on the **Components** tab in the design panel.

You have to define this parameter whether the variables in the workbook are merged or not.

- **Source**

Enter one of the following sources:

- PROMPT to get a list of the prompt members.

If you use PROMPT as source, you have to define the data source alias whether the variables in the workbook are merged or not.

- FILTER to get a list of the dimension members.
- FILTER_MASTERDATA to get a list of all members of the master data for the selected dimension.
- PLAN_PARAMETER to get a list of all variables of a planning object.

- **Name**

Enter the technical name of the dimension or prompt whose members should be listed.

- **Search**

You can enter a search string to restrict the list of members. This parameter is optional.

- **Search in**

Enter if the search should be executed in the KEY or TEXT column. This parameter is optional. The default entry is KEY.

Example:

Cell G15: =SAPListOfMembers("DS_1";"FILTER";"0CALMONTH";"12";"KEY")

Starting in Cell G15, a list is displayed with all members of dimension 0CALMONTH that contain search string 12 in their technical name (KEY).

3.4.19 SAPListOfMessages

This function includes messages into the sheet. You can select which messages, according to their level of severity, should be displayed in the sheet.

The formula consists of 2 parameters and is made up of the following arguments:

- **Severity**

You can enter the following severities: INFORMATION, WARNING, ERROR and CRITICAL.

INFORMATION is the least severe category, CRITICAL the most severe. If you select a severity, all messages are shown which have this severity or higher. If you select WARNING, for example, all messages with severity WARNING, ERROR and CRITICAL are displayed.

- **Show Details**

If you enter FALSE, the messages are listed with severity and text. If you enter TRUE, additional information is displayed, for example message class, ID and Vars for BW system messages.

Example:

Cell H11: =SAPListOfMessages("ERROR";FALSE)

In cell H11, all messages with severity ERROR and CRITICAL are displayed with severity and text.

3.4.20 SAPListOfStaticFilters

This function returns a list of all static filters of a data source. Static filters are defined in the underlying source and cannot be changed by the user.

The formula consists of 2 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.

- **Member Display**

You can enter TEXT or KEY to define how the filtered members should be displayed in the workbook.

Example:

Cell G15: =SAPListOfStaticFilters("DS_1";"TEXT")

If you enter the formula in cell G15, the static filters of data source DS_1 are displayed as text.

3.4.21 SAPListOfVariables

This function returns a list of all variables of a data source or planning object.

The formula consists of 3 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias of the data source or planning object. You can set the alias when configuring the data source or planning object on the **Components** tab in the design panel.

You have to define this parameter if the variables in the workbook are not merged. If the variables in the workbook are merged, you can define this parameter but you do not have to define it.

- **Member Display**

You can enter TEXT or KEY to define how the filtered members should be displayed in the workbook.

- **Display**

Enter one of the following values:

- ALL to display all variables (filled and unfilled) including variables not visible on the prompts dialog.
- PROMPTS to display all variables (filled and unfilled) visible on the prompts dialog.
- ALL_FILLED to display all filled variables including variables not visible on the prompts dialog.
- PROMPTS_FILLED to display all filled variables visible on the prompts dialog.
- PLAN_PARAMETER to display all variables (filled and unfilled) of a planning object.

This parameter is optional. The default value is PROMPTS_FILLED.

Example:

Cell F20: =SAPListOfVariables("DS_2";"TEXT")

If you enter the formula in cell F20, all BW variables with values in data source DS_2 are listed with their values in the worksheet. The first variable name is displayed in cell F20, the next in cell F21 and so on. The corresponding values are listed in G20, G21 and so on.

3.4.22 SAPSetFilterComponent

This function creates a filter component and sets the members selected by the user as a filter. You can click the filter icon to change your filter definition in a dialog box.

This formula consists of 4 parameters and is made up of the following arguments:

- **Data Source**

Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.

- **Dimension Name**

Enter the technical name of the dimension.

- **Target Data Source**

You can enter ALL or a list of formula aliases for data sources that should be affected.

- **Selection Type**

Enter one of the following selection types:

- SINGLE

With this selection type, you can select only one member for filtering.

- MULTIPLE

With this selection type, you can select multiple members for filtering.

- LOWERBOUNDARY

With this selection type, you can define a member as a lower boundary, for example a date.

- UPPERBOUNDARY

With this selection type, you can define a member as an upper boundary, for example a date. You can also insert two filter components in your analysis to define a period of time with a lower boundary date and an upper boundary date.

Example:

Cell E25: =SAPSetFilterComponent("DS_1";"0DIVISION";"ALL";"MULTIPLE")

If you enter the formula in cell E25, the members that are currently filtered are displayed in cell E25. If you select the filter icon next to cell E25, you can change your filter definition.

3.5 Converting crosstab cells to formula

You can convert all cells of a crosstab into formulas with one step. This deletes the crosstab object and defines every row in the table as a Microsoft Excel formula. The result values called from the server with the formula are still displayed in the table. The formula of the selected cell is displayed in the formula bar. In formula mode, you can edit the analysis table using Microsoft Excel formatting and formula functions and make further calculations using the existing data.

In formula mode, you can use all Microsoft Excel formatting functions. With the deletion of the design item, the individual formatting of the data will not be overwritten by the standard formatting in the crosstab the next time you update this data. For example, if you select a color to highlight interim results in the table and then navigate in this table, only the data for the values from the server is called and not the standard formatting from the crosstab. Your individual formatting is retained.

You can use the Microsoft Excel formula functions to make further calculations on the basis of existing data. You can also copy the formula for a cell to another cell outside the table and thus work independently of the original table. If the workbook contains two crosstabs based on different data providers, you can combine the data from both data providers for your calculations.

The Formulas

Formulas with the following functions are composed in the formula mode:

- SAPGetData
- SAPGetMember
- SAPGetDimensionInfo

Examples for working in formula mode

In formula mode, you can use various functions to modify the layout and perform additional calculations.

- You can highlight cells by formatting the font and background color.
- You can insert spaces to make the display easier to read.
- You can copy parts of the table or individual cells to another position in the workbook in order to compare particular values.
- You can re-use cells.
- You can overwrite a members with another one, or add one in order to call data that you need from the BI server. If member "3.2007" is used to read the sales revenue for March 2007, for example, you can replace the 3 with a 4, thus using member "4.2007" to obtain the sales revenue for April 2007, provided that the data provider contains this data.
- You can also calculate additional subtotals.
- You can create offers based on data from various data providers.

Restrictions

Converting to formula mode has the following consequences:

- Navigation using Drag & Drop is no longer possible.
- The context menu is not available.

Related Topics

- [To convert a crosstab to formula](#)

3.5.1 To convert a crosstab to formula

1. Insert a crosstab into a workbook
2. Choose **Convert to Formula**.

This performs the following steps:

- Texts that are not displayed because they occur several times in a column or row, are repeated in each cell automatically to produce valid formulas. You can also execute this step manually by selecting the **Repeat Members** check box for the crosstab on the **Components** tab in the design panel.
- Every cell in the crosstab is defined as a Microsoft Excel formula.

Note:

All currently displayed cells of the crosstab are converted to formula. Cells in a hierarchy that are currently not expanded, are not converted.

- For dimensions and members displayed as text in the crosstab, the key is added to the data source during conversion to formula. This doesn't change the display in the original crosstab. You will only see the added key if you insert the crosstab with the same data source again in your workbook. Then columns and rows are added to display the key.
- Crosstab object is deleted.

All currently displayed cells of the crosstab are converted to a formula using the functions `SAPGetData`, `SAPGetMember` and `SAPGetDimensionInfo`.

Note:

As long as you haven't changed the data in the table, you can go back to analysis mode by choosing **Undo**.

3.6 Working with macros

Analysis contains API methods that can be used in VBA macros. Macros are created in the Visual Basic Editor. The Visual Basic Editor can be used to write and edit a macro that is attached to a Microsoft Office Excel workbook. The macros can be connected to UI elements that are available on the **Developer** tab in the menu.

Note:

In Microsoft Excel 2003, you can find the UI elements in the **Forms** toolbar.

You can also use the Analysis functions for creating formulas in VBA macros.

Microsoft Office documentation provides information about creating and using VBA macros. The following section describes the Analysis API methods.

The following methods are available in Analysis:

- `SAPAddMessage`
- `SAPCallMemberSelector`
- `SAPExecuteCommand`
- `SAPExecutePlanningFunction`
- `SAPExecutePlanningSequence`
- `SAPGetCellInfo`
- `SAPGetProperty`
- `SAPMoveDimension`
- `SAPSetFilter`
- `SAPSetRefreshBehaviour`
- `SAPSetVariable`

3.6.1 SAPAddMessage

With this API method, you can define messages and add them to the standard message dialog.

To call the method, use Application.Run and specify the following input parameters:

- **Message**
Enter the message text.
- **Severity**
You can enter one of the following severities: INFORMATION, WARNING, ERROR or CRITICAL. This parameter is optional. The default value is INFORMATION.
- **Details**
You can enter a long text with additional information to the message. This parameter is optional.

The system returns one of the following output parameters for each function execution:

- 0 = execution failed.
- 1 = execution successful.

Example:

```
Dim IResult As Long
```

```
IResult= Application.Run("SAPAddMessage", "This is a new error message!", "ERROR")
```

The message 'This is a new error message' with severity Error is displayed in the message dialog.

3.6.2 SAPCallMemberSelector

With this API method, you can call the standard input help (member selector) from the worksheet. The standard input help is the dialog that is used for filtering in Analysis. If you define the ReferenceCell parameter, the inplace selector is called. The inplace selector is a smaller version of the member selector. You can use it to select members of a dimension, but you cannot change how members are displayed in the selector, for example to change the display from key to text.

To call the method, use Application.Run and specify the following input parameters:

- **Formula Alias**
Enter the formula alias for the data source or planning object. You can set the alias when configuring the data source on the **Components** tab in the design panel.

You have to define this parameter if the variables in the workbook are not merged. If the variables in the workbook are merged, you can define this parameter but you do not have to define it.

- **Selector Type**

Enter one of the following selector types:

- PROMPT to get a selector with the prompt members.
- FILTER to get a selector with the dimension members.
- FILTER_MASTERDATA to get a selector with all members of the master data for the selected dimension.
- PLAN_PARAMETER to get a selector with all variables (filled and unfilled) of a planning objects.

- **Name**

Enter the name of the variable or dimension.

- **ReferenceCell**

Define the cell in which the inplace selector is displayed. This parameter is optional.

The system returns one of the following output parameters for each function execution:

- 0 = execution failed.
- 1 = execution successful.

Example: CallMemberSelectorFilter

```
Dim IResult As Long
```

```
IResult= Application.Run("SAPCallMemberSelector", "DS_1", "FILTER", "0MATERIAL")
```

The member selector for dimension 0MATERIAL is displayed, and you can select the members you need. You can also select a range of members or change how the members are displayed in the selector.

Example: CallMemberSelectorInplace

```
Dim IResult As Long
```

```
IResult= Application.Run("SAPCallMemberSelector", "DS_1", "FILTER", "0MATERIAL", ActiveCell)
```

The inplace selector for dimension 0MATERIAL is displayed in the active cell, and you can select the members you need.

3.6.3 SAPExecuteCommand

You can use this API method as a generic command to execute the commands described below.

The method supports the following commands:

- **PlanDataTransfer**
Use this command to recalculate entered planning data.
- **PlanDataSave**
Use this command to save entered planning data.
- **PlanDataReset**
Use this command to reset planning data entered since the last save of data.
- **PlanDataClientReset**
Use this command to reset planning data entered since the last successful recalculation of data.
- **PlanDataToDisplayMode**
Use this command to quit input-ready mode and switch to display mode while planning data.
- **PlanDataToChangeMode**
Use this command to switch to input-ready mode while planning data.
- **PauseVariableSubmit**
Use this command to pause the execution of variables using the parameter On. To end the pausing, use the command with parameter Off.

When **PauseVariableSubmit** is in mode "On", all Analysis formulas and most of the Analysis macros will not be executed. The following macros will be executed: **SAPAddMessage**, **SAPSetVariabel**, **PauseVariableSubmit "Off"**.
- **Hide/Show**
Use this command to hide and show Analysis UI components for a workbook. You can use the command for the types "Ribbon" and "ContextMenu".
- **Refresh**
Use this command to initially refresh the data in the workbook. You can specify one data source or one planning object as a parameter that should be refreshed. If you do not enter a parameter or you enter the string 'ALL' as parameter, all data sources and planning objects will be refreshed. If you execute this command for a data source which is already refreshed, all corresponding crosstabs will be redrawn.
- **Restart**
Use this command to restart all or defined data sources in the workbook. If you do not enter a parameter or you enter the string 'ALL' as parameter, all data sources will be restarted. If you want to define a set of data sources that should be restarted, you can enter them as parameter with a semicolon separated list. Restarting a data source has the same effect than deleting and reinserting it. The complete process of adding or initially refreshing a data source is executed. After the restart, new data from the database will be included in the corresponding crosstabs. Before you can restart a data source, the data source has to be initially refreshed, for example with the command **Refresh**.
- **RegisterCallback**

Use this command to register a callback. The following callbacks can be registered with SAPExecuteCommand: AfterRedisplay, BeforePlanDataSave, BeforePlanDataReset.

- UnregisterCallback

Use this command to unregister the callbacks listed above.

The system returns one of the following output parameters for each function execution:

- 0 = execution failed.
- 1 = execution successful.

Example: PlanDataSave

```
Dim IResult As Long
```

```
IResult= Application.Run("SAPExecuteCommand", "PlanDataSave")
```

If you execute this command, the entered planning data is saved.

Example: Refresh one data source

```
Dim IResult As Long
```

```
IResult= Application.Run("SAPExecuteCommand", "Refresh", "DS_1")
```

If you execute this command, the data for data source DS_1 is refreshed.

Example: Refresh all data sources

```
Dim IResult As Long
```

```
IResult= Application.Run("SAPExecuteCommand", "Refresh")
```

If you execute this command, the data of all data sources is refreshed.

Example: Restart

```
Dim IResult As Long
```

```
IResult= Application.Run("SAPExecuteCommand", "Restart", "DS_1"; "DS_2")
```

If you execute this command, the data of data sources DS_1 and DS_2 is refreshed.

Example: UnregisterCallback

```
Dim IResult As Long
```

```
IResult= Application.Run("SAPExecuteCommand", "UnregisterCallback", "AfterRedisplay")
```

If you execute this command, the AfterRedisplay callback is unregistered.

Related Topics

- [Using callbacks](#)
- [Hiding / Showing Analysis UI components](#)
- [SAPSetVariable](#)

3.6.4 SAPExecutePlanningFunction

With this API method, you can execute a planning function. Planning functions are maintained in the **Components** tab in the design panel.

To call the method, use Application.Run and specify the following input parameters:

- **Planning Function Alias**

Enter the formula alias of the planning function. You can set the alias when configuring the planning function on the **Components** tab in the design panel.

The system returns one of the following output parameters for each function execution:

- 0 = execution failed.
- 1 = execution successful.

Example:

Dim IResult As Long

```
IResult= Application.Run("SAPExecutePlanningFunction","PF_1")
```

If you select this command, planning function PF_1 is executed.

3.6.5 SAPExecutePlanningSequence

With this API method, you can execute a planning sequence. Planning sequences are maintained in the **Components** tab in the design panel.

To call the method, use Application.Run and specify the following input parameters:

- **Planning Sequence Alias**

Enter the formula alias of the planning sequence. You can set the alias when configuring the planning sequence on the **Components** tab in the design panel.

The system returns one of the following output parameters for each function execution:

- 0 = execution failed.
- 1 = execution successful.

Example:

```
Dim IResult As Long
```

```
IResult= Application.Run("SAPExecutePlanningSequence","PS_1")
```

If you select this command, planning sequence PS_1 is executed.

3.6.6 SAPGetCellInfo

With this API method, you can define a command to get information on a crosstab cell. You can define the commands for a selection and for a dimension.

For a Dimension call, the result contains the data source alias and the technical name of the dimension.

For a Selection call, the result contains a list of dimensions. Each list entry contains the data source alias, the technical name of the dimension and the technical name of the member. For measures, a list entry contains the data source alias, the measure structure and the technical name of the measure.

To call the method, use Application.Run and specify the following input parameters:

- **Cell**

Enter the cell information for one cell of the worksheet. If the currently selected cell should be evaluated, enter ActiveCell.

If you have selected a range containing more than one cell, the method returns the cell information for the upper left cell in the range.

- **Property**

Enter one of the following properties: SELECTION, DIMENSION, DATASOURCE or CROSSTAB.

- **Filter**

For the SELECTION property, you can enter a list of dimensions to be evaluated, using semicolons as separator. If you do not enter any dimensions, all available dimensions are evaluated.

For DIMENSION, DATASOURCE or CROSSTAB, you do not have to enter a value.

Tip:

If you want to include a check in your program to check if the selected cell is part of the crosstab, you can use the named range of the crosstab as reference. You can find the named range in the Name box in the formula bar. You can modify it on the **Components** tab in the design panel.

Example: Dimension

Dim IResult As Variant

```
IResult= Application.Run("SAPGetCellInfo", ActiveCell, "DIMENSION")
```

If you select a dimension cell in the crosstab and execute the command, it returns the data source alias and the technical name of the dimension.

Example: Selection

Dim IResult As Variant

```
IResult= Application.Run("SAPGetCellInfo", ActiveCell, "SELECTION")
```

In your current analysis, you have filtered the dimensions 0MATERIAL and 0COUNTRY in rows and the measure 0SALESVOLUME in columns. If you select a data cell and execute the command, it returns a list with three rows: One with the measure information, one with the information for dimension 0MATERIAL and one with the information for dimension 0COUNTRY.

Related Topics

- [The Components tab](#)

3.6.7 SAPGetProperty

You can use this API method to read properties of a data source or a workbook. The command returns a boolean value: True or False.

To call the method, use Application.Run and specify the following input parameters:

- **Property Name**

Enter one of the following properties:

- IsDataSourceActive to check if a data source is active.
- IsDataSourceEditable to check if the data source is input-ready.
- HasChangedPlanData to check if the workbook contains changed planning data.
- IsAutoRefresh to check if the automatic refresh is active or paused.
- IsConnected to check if a data source is already connected.
- LastError to get information on an error that occurred while executing Analysis API methods and functions.

- **Data Source Alias**

Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel. This parameter is optional.

If you use the `HasChangedPlanData` or `LastError` property, you should not enter a data source alias, because this property is not related to a specific data source but to the complete workbook.

Example:

```
Dim IResult As Variant
```

```
IResult= Application.Run("SAPGetProperty", "IsDataSourceEditable", "DS_1")
```

If you execute the command, it returns the boolean value. If the data source DS_1 is editable, the value is True.

Example:

```
Iret = Application.Run("SAPGetProperty", "IsConnected", "DS_1")
```

If you execute the command, it returns the boolean value. If the data source DS_1 is connected, the value is True. |

Related Topics

- [Using LastError information](#)

3.6.8 SAPLogon

You can use this API method to trigger a logon to a system for a specified data source.

To call the method, use `Application.Run` and specify the following input parameters:

- **Formula Alias**
Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.
- **Client**
Enter the system client. This parameter is only relevant for logon to an SAP system.
- **User**
Enter the user.
- **Password**
Enter the password.
- **Language**
Enter the logon language. This parameter is optional.

Example:

```
Iret = Application.Run("SAPLogon", "DS_1", "000", "Anzeiger", "Display")
```

If you execute the command, the system of data source DS_1 with client 000 will be connected for user Anzeiger with password Display.

3.6.9 SAPMoveDimension

With this API method, you can define the position of a dimension in the crosstab, swap the dimension with another one or position a dimension relative to another one.

To call the method, use Application.Run and specify the following input parameters:

- **Data Source Alias**

Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.

- **Dimension Key**

Enter the technical name of the dimension.

- **PositionBy**

Enter one of the following constants:

- ROWS to move a dimension to the rows.
- COLUMNS to move a dimension to the columns.
- FILTER to move a dimension to the background filter.
- AFTER to move a dimension after a relative dimension.
- BEFORE to move a dimension before a relative dimension.
- SWAP to swap a dimension with a relative dimension.

- **Identifier**

If PositionBy is set to AFTER, BEFORE or SWAP, enter the technical name of the relative dimension.

If PositionBy is set to ROWS, COLUMNS or FILTER, enter the position as an index. The index is 1-based and optional. If the index is not valid, the command will not be executed. If no index is specified, the dimension is added at the end.

The system returns one of the following output parameters for each function execution:

- 0 = execution failed.
- 1 = execution successful.

Example:

Dim IResult As Long

```
IResult= Application.Run("SAPMoveDimension", "DS_1", "0COUNTRY", "AFTER", "0CALMONTH" )
```

If you execute this command, dimension 0COUNTRY will be inserted in the crosstab to the position after dimension 0CALMONTH.

Example:

Dim IResult As Long

```
IResult= Application.Run("SAPMoveDimension", "DS_1", "0COUNTRY", "ROWS", "2")
```

If you execute this command, dimension 0COUNTRY is added as second entry to the rows in the crosstab.

3.6.10 SAPSetFilter

With this API method, you can define which members of a dimension should be filtered. You can also use this method to define the filter for a planning function.

To call the method, use Application.Run and specify the following input parameters:

- **Formula Alias**
Enter the formula alias for the data source or planning function. You can set the alias when configuring the data source on the **Components** tab in the design panel.
- **Dimension**
Technical name of the dimension to be filtered.
- **Member**
String that represents the member filter for the dimension, for example technical names or a variable. The "ALLMEMBERS" string or an empty string clears the filter and selects all members. Note the syntax rules for entering values.
- **Member Format**
 - Text
Single member as text.
 - Key
Single member as key.
 - INTERNAL_KEY

- Single member with its internal key.
- INPUT_STRING
 - Complex selection of members.

This parameter is optional. The default value is INPUT_STRING.

Note:

The KEY and INTERNAL_KEY depend on the InfoObject modeling in SAP NetWeaver BW.

The system returns one of the following output parameters for each function execution:

- 0 = execution failed.
- 1 = execution successful.

Example:

Dim IResult as long

```
IResult= Application.Run("SAPSetFilter", "DS_1", "0SOLD_TO__0COUNTRY", "CA;US;DE",  
"INPUT_STRING")
```

With this example, you set the filter for dimension 0SOLD_TO__0COUNTRY of data source DS_1 to the countries USA, Canada and Germany using member format INPUT_STRING.

Related Topics

- [Syntax for entering values](#)

3.6.11 SAPSetRefreshBehaviour

With this API method, you can use different API methods one after the other without the result set being refreshed and without the message dialog being shown.

To call the method, use Application.Run and specify the following input parameters:

- **Mode**
 - Enter if the mode is **On** or **Off**.
- **Message Display**
 - Enter, if the message display is **On** or **Off**. The default setting is **Off**.

The system returns one of the following output parameters for each function execution:

- 0 = execution failed.
- 1 = execution successful.

Example:

```
Dim lResult As Long
Sub SetGermanyAndJuice
    Application.Run("SAPSetRefreshBehaviour","Off")
    Application.Run("SAPSetVariable","OBWVC_COUNTRY","DE")
    Application.Run("SAPSetVariable","OBWVC_PLINE","PL001")
    Application.Run("SAPSetRefreshBehaviour","On")
end
```

Related Topics

- [Syntax for entering values](#)

3.6.12 SAPSetVariable

With this API method, you can define values for input-ready BW variables (prompts). If you want to set multiple variables, you can use the `PauseVariableSubmit` command with the `SAPExecuteCommand` method.

To call the method, use `Application.Run` and specify the following input parameters:

- **Prompt Name**
Name or technical name of the BW variable to be filtered.
- **Prompt Value**
String that represents the value for the prompt, for example the technical name. Note the syntax rules for entering values.
- **Value Format**
 - Text
Single member as text.
 - Key
Single member as key.
 - INTERNAL_KEY
Single member with its internal key.
 - INPUT_STRING
Complex selection of members.
 - **Formula Alias**
Enter the formula alias for the data source. You can set the alias when configuring the data source on the **Components** tab in the design panel.

You have to define this parameter if the variables in the workbook are not merged. If the variables in the workbook are merged, you can define this parameter but you do not have to define it.

Note:

The KEY and INTERNAL_KEY depend on the InfoObject modeling in SAP NetWeaver BW.

The system returns one of the following output parameters for each function execution:

- 0 = execution failed.
- 1 = execution successful.

Example:

Dim IResult As Long

IResult=Application.Run("SAPSetVariable", "0BWVC_COUNTRY", "DE")

With this example, you set the variable for dimension 0BWVC_COUNTRY to country Germany.

Example: Setting multiple variables

```
Call Application.Run("SAPSetRefreshBehaviour", "Off")
Call Application.Run("SAPExecuteCommand", "PauseVariableSubmit", "On")
Call Application.Run("SAPSetVariable", "0BWVC_COUNTRY", "US")
Call Application.Run("SAPSetVariable", "0BWVC_MATERIAL", "Hardware")
Call Application.Run("SAPExecuteCommand", "PauseVariableSubmit", "Off")
Call Application.Run("SAPSetRefreshBehaviour", "On")
```

With this example, you set the variable for dimension 0BWVC_COUNTRY to country USA and the variable for dimension 0BWVC_MATERIAL to material Hardware. Both variables will be executed at once after setting off the command PauseVariableSubmit.

Related Topics

- [Syntax for entering values](#)
- [SAPExecuteCommand](#)

3.6.13 Using Analysis functions

You can use the Analysis functions in VBA macros. The return value of a formula in a macro corresponds to the return value of the formula used in a Microsoft Excel sheet. Depending on the function that you use, a formula can either return a single value (single string value) or a list (array). If the list contains only one line, the returning array is one-dimensional. If the list contains two or more lines the array is two-dimensional.

If you use a formula in a macro, the function is the first parameter followed by the arguments you use to create a formula with this function.

If a formula is invalid, Microsoft Excel returns an error.

Example: Formula returning a single string value

```
Dim IResult As Variant
```

```
IResult = Application.Run("SAPGetVariable", "DS_2", "0BW_VAR", "Value")
```

This formula returns the current value of variable 0BW_VAR.

Example: Formula returning an array

```
Dim IResult As Variant
```

```
IResult = Application.Run("SAPListOfDimensions", "DS_1")
```

This formula returns a list with the dimensions of data source DS_1. If the data source contains only one dimension, the returned array is one-dimensional.

To make your programming easier, you can create an additional function to ensure that the array is always two-dimensional, for example the function GetAsTwoDimArray.

```
Function GetAsTwoDimArray(value As Variant) As Variant
'value is error return the error
If IsError(value) Then
GetAsTwoDimArray = value

'value is array
ElseIf IsArray(value) Then
'first check if the array is two-dimensional
'by requesting the upper bound of the 2nd dimension.
'if this is not the case an error occurs (Err.Number <> 0).

'ignore errors, handled locally
On Error Resume Next
Dim lIndex As Integer
Dim lErrorCode As Integer
lIndex = UBound(value,2)
lErrorCode = Err.Number
'set error handling back to default
On Error GoTo 0

If lErrorCode = 0 Then
'no error: array is two-dimensional
GetAsTwoDimArray = value
Else
'copy one-dimensional array into a two-dimensional one
Dim i As Integer
Dim lArray() As Variant
ReDim lArray(1 To 1, 1 To UBound(value))
For i = 1 To UBound(lArray, 2)
lArray(1, i) = value(i)
Next
GetAsTwoDimArray = lArray
End If

Else
'return Empty
GetAsTwoDimArray = Empty

End If

End Function
```

In the following example, a list of all dimensions is returned from function `SAPListOfDimensions`. All dimensions which are neither on the rows nor on the columns axis are added to a string that is displayed in the standard message dialog with the API method `SAPAddMessage`. Using the function `GetAsTwoDimArray`, you ensure that the returned array is always two-dimensional.

```
Sub ShowDimensionsNotOnRowsOrColumns()  
Dim lList As String  
Dim lResult As Variant  
lResult = Application.Run("SAPListOfDimensions", "DS_1")  
lResult = GetAsTwoDimArray(lResult)  
  
For i = 1 To UBound(lResult, 2)  
If lResult(i, 3) <> "ROWS" And lResult(i, 3) <> "COLUMNS" Then  
lList = lList & " " & lResult(i, 2)  
End If  
Next i  
  
Call Application.Run("SAPAddMessage", "Dimensions:" & lList, "INFORMATION")  
End Sub
```

Related Topics

- [Working with formulas](#)

3.6.14 Hiding / Showing Analysis UI components

You can use the commands `Hide` and `Show` of the API method `SAPExecuteCommand` to hide and show Analysis UI components.

The changed visibility of UI components is valid for the active workbook in which the VBA macro is executed. If the Analysis tab is hidden in a workbook and the user switches to another workbook, the Analysis tab is visible again. When switching back to the first workbook, the Analysis tab is hidden again.

The following types and IDs are supported:

- Ribbon

- All

Use this ID to hide the Analysis tab in the ribbon.

Example: `Application.Run("SAPExecuteCommand", "Hide", "Ribbon", "All")`

- Default

Use this ID to show the Analysis tab in the ribbon.

Example: `Application.Run("SAPExecuteCommand", "Show", "Ribbon", "Default")`

- ContextMenu

- All

Use this ID to hide the Analysis context menu.

Example: `Application.Run("SAPExecuteCommand", "Hide", "ContextMenu", "All")`

- Refresh_Single

Use this ID to hide the single context menu entry Refresh.

Example: `Application.Run("SAPExecuteCommand", "Hide", "ContextMenu", "Refresh_Single")`

- Default

Use this ID to show the complete context menu.

Example: `Application.Run("SAPExecuteCommand", "Show", "ContextMenu", "Default")`

Related Topics

- [SAPExecuteCommand](#)
- [Enhancing the Analysis Ribbon](#)

3.6.15 Enhancing the Analysis Ribbon

The Microsoft Office Ribbons can be enhanced and adjusted. You can learn more about the general ribbon enhancement in the Microsoft documentation.

If you build an application related to Analysis, you can visualize the dependencies to Analysis by locating the new ribbon tab beside (before/after) the Analysis tab or by adding new ribbon groups to the Analysis tab. The ribbon is described via an XML definition where an identifier is assigned to each item. The Analysis tab has a qualified identifier (idQ) with the name space `SBOP.AdvancedAnalysis.Addin.1.` and the id `com.sap.ip.bi.analysis.menu`.

Example: Adding a tab before the Analysis tab

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui"
xmlns:x="SBOP.AdvancedAnalysis.Addin.1">
<ribbon>
<tabs>
<tab id="myCustomTab" label="Custom Tab" insertBeforeQ="x:com.sap.ip.bi.analysis.menu" >
<group id="myCustomGroup" label="Custom Group" >
<button id="myButton" label="my Button" onAction="myAction" />
</group>
</tab>
</tabs>
</ribbon>
</customUI>
```

With this example, the new tab 'Custom Tab' is added before the Analysis tab containing the group 'Custom Group' with the button 'my Button'.

Example: Adding a ribbon group to the Analysis tab

```
<customUI xmlns="http://schemas.microsoft.com/office/2006/01/customui"
xmlns:x="SBOP.AdvancedAnalysis.Addin.1">
<ribbon>
<tabs>
<tab idQ="x:com.sap.ip.bi.analysis.menu">
<group id="myCustomGroup" label="Custom Group" >
<button id="myButton" label="my Button" onAction="myAction" />
</group>
</tab>
</tabs>
</ribbon>
</customUI>
```

With this example, the group 'Custom Group' with the button 'my Button' is added to the Analysis tab.

Related Topics

- [Hiding / Showing Analysis UI components](#)

3.6.16 Using callbacks

Analysis offers different callbacks that are executed with certain events. Callback `Workbook_SAP_Initialize` will always be executed. The other callbacks listed below must be registered with the API method `SAPExecuteCommand` before being used for the first time. To unregister the callbacks, you can also use `SAPExecuteCommand`.

Every callback can only be registered once in a workbook. If you register one of the callbacks for a second time, the first registration is overwritten.

Note:

Callback `Workbook_SAP_Initialize` has to be defined in the `ThisWorkbook` section of the VBA editor. The other callbacks should be defined in a module. If they are defined in the `ThisWorkbook` or a sheet section, they have to be fully referenced during registration. In the `ThisWorkbook` section, for example, you have to use `ThisWorkbook.<MacroName>` as a reference.

Workbook_SAP_Initialize

Use this callback to define the initialization logic of the workbook. This callback is similar to the `Workbook_Open` event offered by Microsoft Excel, but it is executed after the Microsoft Excel event and after the Analysis standard initialization. If you open a workbook in Microsoft Excel and then activate the Analysis Add-In, the Excel event is executed when opening the workbook, and the Analysis standard initialization and the callback are executed after the activation of Analysis.

The callback has to be defined as a subroutine without input parameters.

Example:

```
' defined in ThisWorkbook
```

```
Public Sub Workbook_SAP_Initialize()  
    ' register callbacks  
    Call Application.Run("SAPExecuteCommand", "RegisterCallback", "AfterRedisplay", "Callback_AfterRedisplay")  
    Call Application.Run("SAPExecuteCommand", "RegisterCallback", "BeforePlanDataSave", "Callback_BeforePlan  
DataSave")  
    Call Application.Run("SAPExecuteCommand", "RegisterCallback", "BeforePlanDataReset", "Callback_BeforePlan  
DataReset")  
End Sub
```

The AfterRedisplay, BeforPlanDataSave and BeforePlanDataReset callbacks are registered during workbook initialization and can therefore be used in the VBA project.

AfterRedisplay

Use this callback to define the logic that should take place every time the workbook is redisplayed. For example, you can automatically adapt the formatting in the workbook after every redisplay.

The callback has to be defined as subroutine without input parameters.

Example:

```
' defined in a module  
Public Sub Callback_AfterRedisplay()  
  
ThisWorkbook.Worksheets("Sheet1").Cells(1, 1).Value = "Last redisplay: "  
ThisWorkbook.Worksheets("Sheet1").Cells(1, 2).Value = Now()  
  
End Sub
```

With this callback, the text "Last redisplay: " is displayed in cell A1 on sheet 1, and in cell A2, date and time of the last redisplay are displayed. The information is updated after every redisplay.

BeforePlanDataSave

Use this callback to define the logic that should be executed every time a user saves plan data. For example, you can execute a planning function (SAPExecutePlanningFunction).

The callback has to be defined as a function returning a boolean value without input parameters. If the function returns false, the save will not be executed.

Example:

```
' defined in a module  
Public Function Callback_BeforePlanDataSave() As Boolean  
  
Dim lResult As Integer  
lResult = Application.Run("SAPExecutePlanningFunction", "PF_1")  
If lResult <> 1 Then  
    ' planning function execution failed, cancel save  
    Call MsgBox("Planning Function (PF_1) execution failed. Data will not be saved.", vbCritical, "Error")  
    Callback_BeforePlanDataSave = False  
Else  
    Callback_BeforePlanDataSave = True  
End If  
  
End Function
```

Every time a user selects the Save button to save planning data, planning function PF_1 is executed. If the planning function can be executed, the planning data is saved. If the execution fails, a message box with the text "Planning Function (PF_1) execution failed. Data will not be saved." is displayed.

BeforePlanDataReset

Use this callback to define the logic that should be executed every time a user resets plan data to the previously saved state. For example, you define that asks the user if the data really should be reset.

The callback has to be defined as a function returning a boolean value without input parameters. If the function returns false, the reset will not be executed.

Example:

```
' defined in a module
Public Function Callback_BeforePlanDataReset() As Boolean

Dim lAnswer As VbMsgBoxResult
lAnswer = MsgBox("Do you really want to reset planning data?", vbYesNo, "Reset")
If lAnswer = vbYes Then
Callback_BeforePlanDataReset = True
Else
Callback_BeforePlanDataReset = False
End If

End Function
```

Every time a user enters planning data and selects **Back > To previous saved state**, a message box with the text ""Do you really want to reset planning data?" is displayed. If the user selects Yes, the entered data is reset to the previous saved state. If the user selects No, the reset is not executed.

Related Topics

- [SAPExecuteCommand](#)
- [SAPExecutePlanningFunction](#)

3.6.17 Using LastError information

You can use the property LastError of the API method SAPGetProperty to get a more detailed error information when executing an Analysis API method or function. The error information contains a number and an English error text. You can enter as second parameter Text or Number if only this information should be returned. If you do not define the second parameter, both values will be returned.

In the table below, you find a list of all errors with number and text.

Number	Text
General errors	

Number	Text
1	A general error occurred.
2	A general error occurred in one of the data sources. This is most probably an error in the backend system or in the communication with the backend system.
3	An error from a previous call is still unhandled.
Errors on pre-execution checks	
10	The BI-Addin is disabled.
11	At least one worksheet is protected.
12	Variable submit is paused.
13	A callback is running.
Parameter errors	
20	A parameter is missing.
21	A parameter has an invalid value.
Analysis object reference errors	
30	No active workbook application.
31	The specified data source is invalid. The alias may be wrong or the datasource is not yet refreshed.
32	The alias cannot be resolved. This applies to macros where an alias may be a data source or a planning object.
33	Variables are unmerged, data source alias needs to be specified.
34	No result set is available for the specified data source.
Excel object reference errors	
40	Invalid sheet reference.
41	Invalid cell reference.

Example:

```
IResult= Application.Run("SAPGetProperty", "LastError", "Number")
```

If you execute this command, it returns '1'.

```
IResult= Application.Run("SAPGetProperty", "LastError", "Text")
```

If you execute this command, it returns the text 'A general error occurred'.

```
IResult= Application.Run("SAPGetProperty", "LastError")
```

If you execute this command, it returns an array with the members { "1", "A general error occurred"}.

Related Topics

- [SAPGetProperty](#)

3.6.18 Syntax for entering values

Follow these syntax rules when you enter members for filtering and values for prompting.

Description	Example
Equal to	15
Exclude value	!22
Value range	1-5
Exclude value range	!6-9
Greater than	>8
Exclude values greater than <value limit>	!>8
Greater than or equal to	>=8
Exclude values greater than or equal to <value limit>	!>=8
Less than	<12
Exclude values less than <value limit>	!<12
Less than or equal to	<=12
Exclude values less than or equal to <value limit>	!<=12
Contains pattern (for example, all values that begin with A)*	A*
Exclude values that contain pattern (for example, exclude all values that begin with A)*	!A*

Description	Example
Dimension hierarchy node	<Dimension Attribute>(<Technical Name of Dimension>) For example: ELEMENT1(WBS_ELEMENT)
Text hierarchy node	<Technical Name of Hierarchy Node>(0HIER_NODE) For example: EUROPE(0HIER_NODE)
Delimiter for multiple values; semicolon followed by a space	1-5; >12; !8
Escape character; backslash without space	\

If you enter a string that contains one of the special characters used in these syntax rules, you can use a backslash as escape character.
For example: Enter the string CA-QU as CA\QU.

*You can not use these syntax rules for filtering.

Analyzing Data

4.1 To open a workbook

You can open a workbook that is stored on the business intelligence platform server. This allows you to see the navigation state that was saved with the workbook. You also get the workbook properties that were saved with the workbook. These properties are part of the **Component** tab in the design panel.

You can navigate through the data and design different views on the data. You can also add new data sources or components.

1. In Microsoft Excel 2007, press the Microsoft Office Button and choose **Open Workbook**.
In Microsoft Excel 2003, choose **Open Workbook** in the Analysis Standard toolbar.
2. Enter your **User**, **Password** and the **WEB Service URL** to the business intelligence platform and press **OK**.
3. Select a workbook.
You can search for a workbook name or select a workbook in the platform folder structure.
4. Press **OK**.
The workbook is opened in Analysis.
5. Choose **Refresh All** to refresh the data sources.

Note:

If the workbook property **Refresh Workbook on Opening** is selected on the **Components** tab in the design panel, the refresh is executed automatically. In this case, you do not need to refresh the data sources manually.

The selected workbook is opened in Analysis and you can start your analysis.

Related Topics

- [Working with Analysis in Microsoft Excel 2007](#)
- [Working with Analysis in Microsoft Excel 2003](#)
- [The Components tab](#)

4.2 Using a default workbook to open a data source

You can define a workbook as the default workbook. The styles, formatting and the workbook settings that you defined in the default workbook are used when you open a data source. If you save the default workbook on a server, you can share it with other users.

If you do not define a default workbook, the data source is opened with the SAP standard settings.

To define a default workbook

1. Open a workbook and insert a data source.
2. Define the styles, formatting and workbook settings that you want to use for your default workbook.

You can insert other elements like charts or formulas for the data source. You can also change the data source alias.

Note:

You can insert more than one data source in a default workbook. The elements and settings from the first data source that you insert, are used as reference when you open a data source with the default workbook. The elements from the other data sources in the workbook are displayed unchanged when you open a new data source with the default workbook.

3. Save the workbook locally or on a server.
4. Choose **Settings** to open the **User Settings** dialog.
5. Define the location of your default workbook.

Select **Browse** to browse to your default workbook.

6. Press **OK**.

The selected workbook is defined as the default workbook.

To open a data source with the default workbook.

1. Open a data source in Analysis.

In Microsoft Office 2010, choose **File > Open > Data Source**.

In Microsoft Office 2007, press the Microsoft Office button and choose **Open > Data Source**.

In Microsoft Excel 2003, choose **Open Data Source** in the Analysis Standard toolbar.

2. Log on to a system.
3. Select a data source.
4. Press **OK**.

A new workbook with the data from the selected data source is inserted in the the workbook using the styles, formatting and workbook settings that you defined in the default workbook for the first inserted data source. All elements that you have defined for the first data source of your workbook are displayed with the data from the new selected data source. If the default workbook contains more than one data

source, the elements from the other data sources are also displayed with the data from the corresponding data sources.

To open a data source without a default workbook

If you do not define a default workbook in the user settings, the data source is opened with the following Analysis default settings:

- The crosstab is inserted starting in cell A1 in the worksheet.
- The default style set is used for formatting. Initially, the SAP style set *SAP Black&White* is set as default.

Related Topics

- [User settings](#)

4.3 Analyzing data with the design panel

The Analysis design panel is an additional element on the user interface that you can use to create new views on your data, to find information on the used data sources and on the components of the workbook. The design panel consists of three tabs: **Analysis**, **Information** and **Components**.

You can show and hide the design panel by choosing **Display** for the design panel. You can modify the size and position of the design panel in Microsoft Excel. Your modifications of the design panel will be preserved even if you close Analysis or hide the design panel and show it again.

You can choose **Pause Refresh** to deactivate the refresh after each navigation step. With the deactivated refresh, you can design a new view on your data with multiple navigation steps. All steps will be executed at once when you choose **Pause Refresh** again to activate the refresh. Note that most functions in the menu are not available when Pause Refresh is activated. You should only work on the Analysis tab when the automatic refresh is deactivated.

Related Topics

- [The Analysis tab](#)
- [The Information tab](#)
- [The Components tab](#)

4.3.1 The Analysis tab

On the **Analysis** tab, you can see the available fields for a single data source and the fields currently used to display the data in a crosstab in columns and rows. If you use multiple data sources in your

analysis, select a crosstab cell of the required data source to specify which data source information should be displayed.

The Analysis tab contains four sections:

- **<Data source>**

The heading for this section is the name of the selected data source. You can check whether you have selected the right data source. The section displays all fields of the data source: measures, dimensions, attributes and hierarchies.

In Bex Query Designer, a dimension can be defined as not to be displayed. These dimensions are displayed in the Analysis design panel in gray font. You can add them to your crosstab and filter them. This will affect the data and result in your crosstab, but the dimensions will not be visible in the crosstab. If you want a dimension to be displayed in an Analysis crosstab, you have to change the definition in BEx Query Designer.

- **Columns**

This section contains all fields that are currently displayed in columns.

- **Rows**

This section contains all fields that are currently displayed in rows.

- **Background Filter**

This section contains all fields that are currently defined as background filters.

Navigating through the data

Using the Analysis tab, you have the following options to analyze data and create new views on it:

- You can use the context menu to move fields in the four sections. You can add a new dimension from the <Data source> section to the Rows, for example.
- You can use drag and drop to move fields in the four sections.
- You can use drag and drop to add rows and columns to the crosstab. Drag a field directly to the crosstab and drop it. The field is added automatically to the corresponding section on the Analysis tab.
- You can also use drag and drop to remove a field from the crosstab. Select the border of a cell in the crosstab to drag the field to the Analysis tab and drop it there.
- You can use the context menu to create, change and remove filters. If a filter is defined for an object, you see a filter icon beside this object.

4.3.2 The Information tab

On the **Information** tab, you can see detailed information about a data source or the complete workbook. You can also find information on filters and variables on this tab. The general information is displayed as text elements.

In the **Information for** list, you can select the complete workbook or one of the inserted data sources. The information fields are displayed for the selected object. You can insert these fields in the analysis using drag and drop.

For a workbook, the following information is displayed:

- Workbook Name
- Created By
- Variables
- Logged On User
- Last Refreshed At
- Created At
- Last Changed At

For a data source, the following information is displayed:

- Data Source Name
- Key Date
- Last Data Update
- Variables
- Filter
- Query Technical Name
- InfoProvider Technical Name
- InfoProvider Name
- Created By
- Last Changed By
- Last Changed At
- System
- Logged On User

Related Topics

- [To insert an info field](#)
- [To insert a filter](#)

4.3.3 The Components tab

On the **Components** tab, you can see a list of all components used in the workbook together with the properties of these components. You can select if the components should be listed by data source or by sheet. In both cases, the highest node of the list is the workbook. Below this, the data sources or the sheets are displayed with their components.

You can use the context menu to manage the components. For more information, see the corresponding document in this guide.

A set of properties is available for each component. To see and change the properties for a component, select the component in the structure.

The following table lists the available components and properties:

Component	Properties	Description
Workbook	Name	Displays the name of the workbook. The workbook name is defined when the workbook is saved.
	Refresh Workbook on Opening*	<p>If you select this check box, the data sources in the workbook are refreshed every time the workbook is opened.</p> <p>If this check box is not selected, the data in the workbook is not automatically refreshed on opening. You can refresh the data sources manually by choosing Refresh All in the menu.</p>
	Force Prompt for Initial Refresh*	If you select this check box, the prompting dialog is displayed on every refresh.
	Store Prompts with Workbook*	If you select this check box, the defined prompt values are saved with the workbook.
	Remove Data Before Saving	If you select this check box, the workbook is saved without the data. When you reopen the workbook, no data is displayed. To display the data, refresh the data sources manually by choosing Refresh All in the menu.
	Merge Variables*	<p>If you select this check box, the variables are merged for all data sources.</p> <p>If the check box is not selected, the variables can be defined for each data source separately.</p> <p>Note: The following functions and API methods might need different parameters depending on the selection for merging variables:</p> <ul style="list-style-type: none"> • SAPListOfMembers • SAPListOfVariables • SAPCallMemberSelector • SAPSetVariable
	Planning: System for Data Write Back	

Component	Properties	Description
		<p>Displays the currently selected planning system.</p> <p>For an empty workbook, this property is empty.</p> <p>When the first data source is inserted, the corresponding system is visible as the planning system.</p> <p>When you insert data sources from different systems, the system of the first inserted data source is the planning system. You can change the planning system by selecting another system in the list. The changed property value will only become effective when you close and reopen the workbook.</p> <p>When you change the planning system in a workbook before you are connected to a server, the changed property will become effective when you connect to a server. In this case, you do not need to close and reopen the workbook again.</p>
Data source	Data Source Name	Displays the data source name. You can change it here.
	Data Source Technical Name	Displays the technical name of the data source.
	Formula Alias	Displays the formula alias. You can change it here.
	System Alias	Displays the system alias.
	Planning: Open Data Source Input-Enabled	<p>If you select this check box, the input-ready data source is opened in change mode when you open the workbook, and you can immediately start entering planning data.</p> <p>If you do not select this check box for an input-ready data source, you cannot enter and save planning data for this data source until you have selected the check box.</p>
Sheet	Name	Displays the sheet name.

Component	Properties	Description
Crosstab	Name	Displays the crosstab name. You can change it here.
	Formula Alias	<p>Displays the formula alias of the crosstab. You can change it here.</p> <p>The crosstab formula alias is the second part of the named range for a crosstab. The first part is always SAP. For example, if the crosstab formula alias is CrossTab1, the named range for this crosstab would be SAPCrossTab1. You can select a crosstab with its named range in the Name box in the formula bar.</p>
	Data Source Name	Displays the data source name and the data source alias.
	Range	Displays the cell range of the crosstab in the sheet. You can change the range to move the crosstab in the sheet.
	New Lines	You can define how many new lines should be added to the crosstab for planning data. The default number is 5 lines. The new lines are only visible in change mode.
	Apply Default Formats	If you select this check box, the default formats are applied after each navigation step.
	Display Symbols for Parent Members	If you select this check box, a symbol is displayed for parent members in a hierarchy (+/-).
	Optimum Cell Width/Height	If you select this check box, the table is displayed with optimum cell width and height.
	Repeat Members	If you select this check box, texts that are not displayed because they occur several times in a column or row are repeated in each cell.
	Wrap Headers	<p>In BEx Query Designer, you can define if key figures should be displayed in multiple lines.</p> <p>If you select this check box, these key figures/measures are displayed in Analysis as defined in Query Designer. If you clear the check box, the measures are displayed in one single line.</p>

Component	Properties	Description
Planning Objects	Name of Planning Function / Sequence	Displays the name of the planning object.
	Technical name of Planning Function / Sequence	Displays the technical name of the planning object.
	Alias	Displays the alias of the planning object. You can change it here.
Filter	Data Source Name	Displays data source name.
	Dimension	Displays the name of the filtered dimension.
	Filter applies to the following data sources	You can select which of the data sources in the workbook should be filtered.
Chart	Name	Displays sheet and chart name.
	Cross Tab Name	Displays the crosstab name.
	Sheet	Displays the sheet name. You can change it here.

* For more information on workbook properties, see [To select workbook properties for prompting.](#)

Related Topics

- [Managing components in the design panel](#)

4.3.4 Managing components in the design panel

On the **Components** tab, all components included in a workbook are displayed in a tree-like structure. You can select if the components should be listed by data source or by sheet. In both cases, the highest node of the list is the workbook. Below this, the data sources or the sheets are displayed with their components.

You can use the context menu to manage the components. The following sections describe the functions of the context menu.

To insert a component

On workbook level, you can insert the following components:

- data sources ('Use Data Source')
- planning functions ('Use Planning Function')

- planning sequences ('Use Planning Sequence')

On data source level, you can insert a crosstab for this data source ('Insert Crosstab'). A dialog box appears to enter the sheet and cell information where the copied crosstab should be inserted.

On the planning objects level, you can insert planning functions ('Use Planning Function') and planning sequences ('Use Planning Sequence')

To refresh a component

You can refresh the following components with the context menu:

- workbooks
- crosstabs
- charts

To copy a component

You can copy the following components with the context menu:

- data sources

When you copy a data source, the copied data source is added to the tree-like structure with a new data source alias. No crosstab or other component of the original data source is added to the workbook.

- crosstabs

When you copy a crosstab, a dialog box appears to enter the sheet and cell information where the copied crosstab should be inserted. The new crosstab is inserted on the defined position and in the structure, a entry with data source, data source alias and crosstab is added.

- sheets

When you copy a sheet, the copied sheet is added to the tree-like structure.

To move a component

You can move crosstabs and charts ('Move To'). A dialog box appears to enter the sheet and cell information where the crosstab or chart should be moved to.

To jump to a component

You can use the context menu to jump to a crosstab or a filter ('Jump To'). You can use this function to jump to a crosstab on another sheet in the workbook, for example. The sheet with the selected crosstab will be displayed.

To execute planning objects

You can use the context menu to execute planning functions and planning sequences.

To delete a component

You can delete the following components with the context menu:

- data sources
- crosstabs
- charts

- filters
- planning functions
- planning sequences

4.4 Prompting

In the prompting dialog, you can set values for prompts. Prompts are defined as variables in BEx Query Designer. The variables are parameters of a BW query and are filled with values when you insert a query as a data source in a worksheet. They serve as placeholders for members, hierarchies, hierarchy nodes, formula elements and texts. After defining the variable values, the crosstab is displayed according to the selected values.

To be editable in the prompting dialog, a variable must be defined as input-ready and set to Manual Input in Query Designer. A variable can be defined as mandatory or optional in Query Designer.

If a query contains mandatory variables without default values, the prompting dialog appears automatically when you insert the query in a worksheet and requires the definition of mandatory variable values before the crosstab is added. If you cancel the prompting dialog, the query data is not included in the workbook.

For queries with optional variables only, you can open the prompting dialog with the prompting icon in the menu and define variable values. You can also use this manual option to open the dialog to change existing values.

At workbook level, you can select properties to define how the prompting dialog and defined values behave in workbooks. For example, you can define that variable values are saved in a workbook. If you open this workbook again, the data is displayed according to the defined values. You can then open the prompting dialog and change the values.

If there are no input-ready variables in the query that you use as your data source, the prompting dialog is disabled.

Note:

Objects that are defined as input-ready variables in BEx Query Designer are called prompts in Analysis, edition for Microsoft Office. In the following documentation the Analysis term **prompt** is used for these variables. A query can also contain other variable types. All variables and values of these variables that a query contains, are displayed in the **Variables** area on the **Information** tab in the design panel.

To find more information about the definition of variables in the BEx Query Designer, go to the SAP Help Portal at "<http://help.sap.com>" **SAP NetWeaver > Select release and language > SAP NetWeaver Library > SAP NetWeaver by Key Capability > Business Intelligence > BI Suite: Business Explorer > Query Design: BEx Query Designer > Reusable Query components > Variables.**

Related Topics

- [To define prompt values](#)
- [To select workbook properties for prompting](#)

4.4.1 To define prompt values

1. Open the **Prompts** dialog.

In the **Prompts Summary** area, you see a list of all available prompts in the query and the selected values. In the **Specify Value for Prompts** area, you can expand single prompts to define values. The mandatory prompts are marked with an asterisk. If default values for the prompts are defined in BEx Query Designer, they are displayed as selected in the prompting dialog.

If you have selected to not merge the variables for this workbook on the **Components** tab in the design panel, the prompts are displayed separately for each data source and you can define different values for the same prompt for the corresponding data sources.

2. In the list box, select whether all prompts, no prompts or only the mandatory prompts are expanded in the **Specify Value for Prompts area**:

- Select **Hide All** if all prompts should be displayed collapsed.
- Select **Show All** if all prompts should be displayed expanded. This is the default option.
- Select **Hide Optional Prompts** if only the mandatory prompts should be displayed expanded.

3. Define values for the expanded prompts.

You can define values for the following prompt types. Depending on the prompt definition in BEx Query Designer, you have various options:

Prompt Type	Value Definition
Dimension*	<p>Single value</p> <p>You can select one member as the prompt value for this dimension.</p>
Dimension*	<p>Multiple value</p> <p>You can select multiple members as the prompt values for this dimension.</p> <p>Press the + button, to add a field for additional values. Press the Filter button if you want to select multiple values in a list at once. To remove a selected member, press the red X button.</p>
Dimension*	<p>Value range</p> <p>You can select a start and an end member to define a range of values for this dimension.</p>
Dimension*	<p>Selection option</p> <p>You can select an operator and corresponding members to define a selection for this dimension. The following operators are available:</p> <ul style="list-style-type: none"> • = Equal To • != Not Equal to • >= Greater Than or Equal To • > Greater Than • < Less Than • <= Less Than or Equal To • [] Between • ![] Not Between <p>You can add multiple selections for this dimension. Press the + button, to add a selection. To remove a selection, press the red X button.</p> <p>Definitions with other operators that are available in BEx Query Designer 3.5, can be displayed and deleted but not edited.</p>
Hierarchy	<p>You can select a hierarchy from the list box to define a prompt value.</p>
Hierarchy node*	<p>Single value</p> <p>You can select one hierarchy node as the prompt value for the selected hierarchy.</p>

Prompt Type	Value Definition
Hierarchy node*	<p>Multiple value</p> <p>You can select multiple hierarchy nodes as the prompt values for the selected hierarchy.</p> <p>Press the + button, to add a field for additional value. Press the Filter button if you want to select multiple values in a list at once. To remove a selected member, press the red X button.</p>
Formula	<p>You can enter a numeric value.</p> <p>The numeric value is used for measure calculations defined in BEx Query Designer with prompts.</p>
Text	<p>You can enter a text.</p> <p>This prompt type can be used to assign text to columns or row headers or to change the description of a calculated measure.</p>

* For these prompt types, you can enter the values manually or using input help. When you open input help, no values are selected. You can search for members or select members directly from the list. You can also define if the members in the list should be displayed with key, text, or both. For hierarchical dimensions, you can define if the members should be displayed in hierarchies, at leaf level, or on a specific level.

4. Press **OK** to apply the selected values to your data.

The system validates all prompt values. If the validation is successful, the crosstab is displayed according to your selection.

The data in the worksheet is displayed according to your selection.

If you select a data source on the **Information** tab in the design panel, you can see the defined values in the **Variables** area. If you select the workbook on the **Information** tab in the design panel, you can see the defined values in the **Variables** area if the variables in the workbook are merged. If the variables are not merged, the message 'Variables are not merged' is displayed in the **Variables** areas.

The defined prompt values can be saved with the workbook and are then available to other users or in other sessions. To change the prompt values, open the prompting dialog again.

Related Topics

- [To select workbook properties for prompting](#)

4.4.2 To select workbook properties for prompting

You can select various workbook properties in the design panel that affect the behavior of the prompting dialog and existing prompt values:

1. Open the **Design Panel** in the menu.
2. Select the **Components** tab.
3. Select the required workbook properties in the **Properties** area. The following properties are available:

- **Refresh Workbook on Opening**

If you select this option, the data sources in the workbook are refreshed every time the workbook is opened. The behavior of the prompting dialog depends on the other properties you selected for this workbook.

If this check box is not selected, the data in the workbook is not automatically refreshed on opening. You can refresh the data sources manually with choosing **Refresh All** in the menu.

- **Force Prompt for Initial Refresh**

If you select this option, the prompting dialog is displayed on every refresh. The dialog appears, even if the query contains only optional prompts. If values are already saved with the workbook, these are displayed as predefined settings in the dialog. You can accept or change these values. A refresh happens when you open a workbook and select refresh, when you select the setting above and open a workbook or when you insert a new data source.

If this property is not selected, the system checks whether the workbook contains mandatory variables that no values are available for. If no values are available, the prompting dialog is displayed; if values are available, the workbook is displayed straight away.

- **Store Prompts with Workbook**

This is the default property. If it is used, the defined prompt values are saved with the workbook and will be applied to the data when the workbook is refreshed during the current session or closed and opened again. The prompt values are also available to other users using the workbook. To change the values, you can open the prompting dialog manually.

If this property is not selected, the defined values will not be saved with the workbook.

- **Merge Variables**

If you select this option, all prompts of the workbook are listed in the **Prompts** dialog. You can define values for each prompt. These prompt values are valid for all data sources that contain the prompt.

If this property is not selected, the data sources in the workbook are displayed in the **Prompt Summary** area of the **Prompts** dialog and the corresponding prompts are listed in the **Specify Value for Prompts** area. You can define different values for the prompts for each data source.

If you select to merge variables after defining values for each data source separately, the values defined for the data source that was first inserted into the workbook, are used for the merged variables.

The behavior of the prompting dialog and saved prompt values will work according to your workbook properties selection.

4.5 Filtering data

In Analysis, you can define criteria to restrict the data displayed in your analysis to a subset of data. This action is called filtering. With filtering, you define which subset of data appears in the analysis. This allows you to create new, more specific views of your data. Filters are additive, which means that each additional filter is based on the current filter and further reduces the subset of data.

There are two types of filters. With "Filtering members", you can specify which members to display and which not to display in your analysis. You can also define members to use as filters in the background of the analysis. With "Filtering measures", you can specify which measure values should be displayed in the analysis. For example, you can define that you want to see the Top 3 regions in sales volume for each product in your analysis. You can also filter out rows and columns that contain only zeros.

Related Topics

- [Filtering members](#)
- [Filtering measures](#)
- [To show/hide zeros in rows and columns](#)

4.5.1 Filtering members

By selecting and removing members of your analysis, you can create new views of your data. You can filter members of a flat dimension with the filter dialog or directly on the crosstab. You can also filter members of a hierarchical dimension (hierarchy). The measures in an analysis are combined to one dimension, the measure dimension. You can also define criteria that are used in the background of the analysis for filtering.

For example, if you analyse the sales volume of your company in the last three years, and you only want to see the values for the last three month of each year, you can remove the other months from your analysis.

Filtering members is a static action. Members that you have removed from your analysis remain excluded unless you change the filter criteria and include them again.

Member filters affect the totals and subtotals in your analysis. Only the values for the displayed members are included in the totals.

Related Topics

- [To filter data by member](#)
- [To filter data by member within hierarchies](#)
- [To filter members directly on the crosstab](#)
- [To filter out members directly on the crosstab](#)
- [To create a background filter](#)
- [To remove a filter by member](#)

4.5.1.1 To filter data by member

1. Select a cell in the crosstab that belongs to the dimension you want to use for filtering.
2. Choose **Filter > Filter by member**.
The "Filter by Member" dialog box appears.

Note:

You can also open this dialog box using the context menu on the crosstab or the design panel.

3. Select **Individual Selection** if you want to filter individual members of the dimension.
 - a. If you want to search for members, clear the **Select All** check box in the members list and enter a search string in the entry field. In the drop-down list of the search button, you can select if you want to search for a key or text.
Members that match the search string are displayed in the members list and can be selected.
 - b. With the **Display** list, you can select how the members are displayed in the list. The following options are available: Key, Text, Text and key, Key and text.
 - c. Press the **Show Selected** button if only the selected members should be displayed.
The button name changes to **Show All**. Press this, if you want all members to be displayed again.
 - d. Select the members that you want to use for filtering in the members list.
In the members list, all members are selected in the default setting. To deselect a member, clear the corresponding check box. You can also deselect all members by clearing the **Select All** check box and selecting individual members from the list.
 - e. Insert the members that you want to use for filtering with **Paste from Clipboard**.
You can copy a single or a list of members to clipboard and copy your selection to the filter dialog using the **Paste from Clipboard** icon at the left bottom of the filter dialog. The members that you want to copy, can be displayed as either text or key in the list. If you copy a list of members, you can use the following separators: : (colon), ; (semicolon), | (pipe), \t (tab), \r\n (new line).
After pasting from clipboard, the copied members are selected in the member list.
4. Select **Range Selection** if you want to filter for a range of members.

- a. Select an operator.
The following operators are available: Between, Not Between, Greater Than, Greater Than Or Equal To, Less Than, Less Than Or Equal To, Equal To and Not Equal To.
 - b. Select the member range.
For the range definition, the key of the members is relevant, not the text.
 - c. Select **Add Range**.
The range is added to the "Range" area. You can add several ranges to the filter.
5. Press **OK** to apply the filter to your data.

The data in the worksheet is filtered according to your selection. The filter is saved with the workbook and is available to other users or in other sessions. You can also open and edit existing filters again.

4.5.1.2 To filter data by member within hierarchies

The data source you use, includes hierarchies.

1. Select a cell in the crosstab that belongs to the hierarchy you want to use for filtering.
2. Choose **Filter > Filter by member**.

The "Filter by Member" dialog box for hierarchies appears.

Note:

You can also open this dialog box using the context menu on the crosstab or the design panel.

3. If you want to search for members, clear the **Select All** check box in the members list and enter a search string in the entry field. In the drop-down list of the search button, you can select if you want to search for a key or a text.
Members that match the search string are displayed in the members list and can be selected.
4. With the **Display** list, you can select how the members are displayed. The following options are available: Key, Text, Text and key, Key and text and Show Attributes.
In the **Show Attributes** list, you can find the attributes that are available for the members in your hierarchy. For example, a product hierarchy could have size and colour as attributes. You can select and add these attributes to the member list in the filter dialog. This additional information may help you to make the selections for filtering.
5. Select in the list box which members should be displayed in the list for selection.
 - Select **Hierarchy** in the list if you want the members to be listed with the hierarchy.
 - Select **Leaves** if you want all members of all levels to be listed in a flat view.
6. Press the **Show Selected** button if only the selected members should be displayed.
The button name changes to **Show All**. Press this if you want all members to be displayed.
7. Select the members that you want to use for filtering in the members list.

In the members list, all members are selected in the default setting. To deselect a member, clear the corresponding check box. You can also deselect all members by clearing the **Select All** check box and selecting individual members from the list.

If you deselct a node in a hierarchy, all leaf members of this node are deselcted automatically. The parent members of the deselcted node are still selected, but the background color of the check box changes to gray. That shows you that not all leaf members of this node are selected for filtering.

8. Press **OK** to apply the filter to your data.

The data in the worksheet is filtered according to your selection. The filter is saved with the workbook and is available to other users or in other sessions. You can also open and edit existing filters again.

4.5.1.3 To filter members directly on the crosstab

You can filter for one or multiple members directly using the context menu.

1. In the crosstab, select one cell for each member that you want to filter for.
To select multiple members, hold down the Ctrl or Shift key as you select members.
2. Choose **Filter Members** in the context menu.
In the design panel, the filter icon is added to the corresponding dimension.

The data in the worksheet is filtered according to your selection. You can apply this filtering to dimensions with and without hierarchies.

4.5.1.4 To filter out members directly on the crosstab

You can filter out one or multiple members directly using the context menu or drag and drop.

1. In the crosstab, select one cell for each member that you want to filter out.
To select multiple members, hold down the Ctrl or Shift key as you select members.
2. Choose **Filter Other Members** in the context menu.
To filter out the selected members using drag and drop, select the border of the marked field(s) and drag them out of the crosstab.
In the design panel, the filter icon is added to the corresponding dimension.

The data in the worksheet is filtered according to your selection. You can apply this filtering to dimensions with and without hierarchies.

4.5.1.5 To create a background filter

In Analysis, you can set filters for dimensions and hierarchies that you have chosen to be displayed in the crosstab. This allows you to narrow the scope of your analysis to the data you need. You can however also select members for filtering that should not be displayed in the crosstab. These non-visible filters are background filters.

For example, you have selected the dimensions Product and Region to be displayed in the crosstab. The data source contains the sales volume for the years, 2008, 2009 and 2010. If you are only interested in the sales volume for 2010, you can add the Year dimension as a background filter to your analysis and filter for 2010. The analysis will show the data for year 2010, but the Year dimension is not visible in the crosstab.

Note:

Background filters for a measure dimension can contain only one measure as a filter. The same is true for other dimensions that do not aggregate.

You have the following options to add a background filter to your analysis:

1. You can add a dimension to the **Background Filter** area and define a filter:
 - a. Add the dimension that you want to add to the **Background Filter** area in the design panel.
You can add the dimension using drag and drop or you can choose **Move to Background Filter** in the context menu.
 - b. Choose **Filter By Member** in the context menu.
The "Filter by Member" dialog box appears.
 - c. Make your selection and press **OK**.
The background filter is created.
2. Or: You can add a dimension that already has a filter. The existing filter will be applied as a background filter in the crosstab.
3. Or: You can add a background filter using the context menu on the crosstab.
 - a. Select the member(s) that you want to define as background filter in the crosstab.
 - b. Choose **Filter Members and Swap With** in the context menu.
The dimension with the selected filters is added automatically to the **Background Filter** area, and the dimension chosen for swapping with is added to the **Rows** area in the design panel.

The data in the crosstab is displayed according to your filter selection.

4.5.1.6 To remove a filter by member

1. Select a cell in the crosstab that belongs to the dimension you want to remove the filter for.
2. Choose **Filter > Select All Members**.

Note:

You can also choose **Select All Members** in the context menu on the crosstab or the design panel.

All members of the selected dimension are displayed in the crosstab.

4.5.2 Filtering measures

With filtering measures, you can define rules to filter the data of your analysis to your current scope of interest. You can apply one or multiple rules to a measure. Depending on where you open the **Filter by measure** dialog, the filter definition is applied to a selected dimension, to all dimensions or to the most detailed dimension in your analysis. The applied measure filters do not affect totals or subtotals in your analysis.

You can define rules to get a ranked list of data and a list of data matching a threshold value.

In a ranked list filtering, all entries of the currently displayed data are considered and their relationship to each other determines whether an entry is displayed. For example, you can define that the Top 3 entries for a measure should be displayed.

In a threshold list filtering, an entry is filtered independently of the other entries if its value matches the criteria in the defined rule. For example, you can define that all entries are displayed that are greater than a specific threshold value.

Filtering measures is a dynamic action. Whenever you change the view of your data, the filter is applied again. For example, if you add a Top 5 filter, five members are shown in your analysis. If you then add members that you previously removed from the analysis, some of these newly added members could match the Top 5 criteria and replace some of the previously displayed members.

Related Topics

- [To filter data by measure](#)
- [To change a filter by measure](#)
- [To remove a filter by measure](#)

4.5.2.1 To filter data by measure

1. You have the following options to open the filter dialog:
 - Select a cell belonging to a dimension in the crosstab and choose **Filter > Filter by Measure > Edit**.

If you select this option, the filter is applied to the members of the selected dimension.

- Select a cell belonging to a measure in the crosstab and choose **Filter > Filter by Measure > All Dimensions Independently > Edit**.

If you select this option, the filter is applied to all dimensions in the analysis, from the outermost to the most detailed one. For example, in an analysis with three dimensions in columns A, B and C, the filter is first applied to the dimension in column A, then to the dimension in column B and finally to the dimension in column C.

- Select a cell belonging to a measure in the crosstab and choose **Filter > Filter by Measure > Most Detailed Dimension in Rows > Edit**.

If you select this option, the filter is applied to the most detailed dimension of the rows. For example, in an analysis with three dimensions in columns A, B and C, the filter is applied to the dimension in column C. This option is optimized for threshold value filtering.

- Select a cell belonging to a measure in the crosstab and choose **Filter > Filter by Measure > Most Detailed Dimension in Columns > Edit**.

If you select this option, the filter is applied to the most detailed dimension of the columns. For example, in an analysis with three dimensions in rows 2, 3 and 4, the filter is applied to the dimension in row 4. As with the option above, this option is also recommended for threshold value filtering.

The "Filter by Measure" dialog box appears.

Note:

You can also open this dialog box using the context menu on the crosstab or the design panel.

2. In the **Based on measure:** list, select the measure you want to use.
3. In the **Definition** area, select the operator and value for a new rule.

The following operators are available: Between, Not Between, Greater Than, Greater Than or Equal to, Less Than, Less Than or Equal to, Equal to, Top N, Bottom N, Top%, Bottom %, Top Sum, Bottom Sum.

4. Press **Add**.

The new rule is added to the **Rules** area.

5. If you define multiple rules, select the logic for the rules.

Select **AND**, if all rules should be applied simultaneously, or **OR**, if only one of the rules should be applied.

6. Press **OK**.

The data in the worksheet is filtered according to your selection. The filter is saved with the workbook and is available to other users or in other sessions. You can also open and edit existing filters again.

Related Topics

- [To change a filter by measure](#)
- [To remove a filter by measure](#)

4.5.2.2 Operators used for filtering measures

When defining filters by measure, you can create rules to get a list of data for a defined threshold value and rules to get ranked list of data.

To define threshold values, you can use the following operators:

- Between / Not Between

For these operators, you define a range of values that should be included or excluded in the list.

- Greater Than / Less Than

For these operators, you define a specific value to display members that are greater or less than this value.

- Greater Than or Equal to / Less Than or Equal to

For these operators, you define a specific value to display members that are greater or equal to or less or equal to this value.

- Equal to

For this operator, you define a specific value to display members that are equal to this value.

To define a ranked list of values, you can use the following operators:

- Top N / Bottom N

For these operators, you must enter a whole, positive number as the value. The ranked list is arranged according to the defined number.

For example, if you want a list of the three customers with the strongest sales volume and the five customers with with the weakest sales volume, you create a Top 3 and a Bottom 5 rule in the filter definition.

- Top% / Bottom %

For these operators, you must enter a number between 0 and 100. The ranked list is arranged according to the defined percentage.

For example, if you want a list of the strongest products in revenue until 25% of the total revenue is attained, you create a Top 25% rule for this measure with regard to products.

- Top Sum / Bottom Sum

For these operators, you can also enter a floating point number (15.7 for example). The ranked list is arranged according to the defined total value.

For example, you want a list of products with the highest sales volume, whose combined sales volume totals EUR 20,000. All sales volumes are first sorted in descending order. The totals are calculated until the threshold value of 20,000 is exceeded. If the last value that need to be included to achieve EUR 20,000, causes the total to exceed 20,000, it is also included in the list. All products

with a lower sales volume than this product are no longer displayed. Products that exceed the EUR 20,000 threshold remain in the list.

Note:

Do not use these operators for ranked lists for filtering hierarchies.

4.5.2.3 To change a filter by measure

1. Select a cell in the crosstab.

There are four options for defining a filter by measure. To delete an existing filter, you have to select a cell that belongs to the option you used for defining it.

2. Choose **Filter > Filter by Measure > [Selected Filter Option] > Edit**.

Note:

You can also use the context menu on the crosstab.

3. In the **Rules** area, select the **Change** icon in the rule that you want to change.
4. Change the definition of the rule as required and press the **Confirm** icon to change the rule.
5. Press **OK** to apply the changed rule(s) to your data.

The data in the worksheet is displayed according to the changed filter rule(s).

4.5.2.4 To remove a filter by measure

1. Select a cell in the crosstab.

There are four options for defining a filter by measure. To delete an existing filter, you have to select a cell that belongs to the option you used for defining the filter.

2. Choose **Filter > Filter by measure > [Selected Filter Option] > Reset**.

Note:

You can also use the context menu on the crosstab.

The data in the crosstab is displayed without the removed filter.

4.5.3 To show/hide zeros in rows and columns

1. Select a cell in the crosstab.
2. Choose **Filter** and one of the following options:
 - **Suppress Zeros in Rows**
All rows that only contain zeros are removed.
 - **Suppress Zeros in Columns**
All columns that only contain zeros are removed.

The zeros are suppressed in rows and/or columns. When zero suppression is active, a check mark is displayed beside the menu item.

To restore the hidden rows and columns, choose **Filter > Suppress Zeros in Rows** or **Filter > Suppress Zeros in Columns** again.

You can also define how zeros should be displayed in the crosstab if they are not hidden.

Related Topics

- [To define the display of zeros and negative values](#)

4.6 Sorting data

Sorting data is a fundamental part of data analysis. It helps you understand your data better as you can visualize and organize different views on the data in your crosstab. With sorting the data, you are able to quickly identify the data you need for your decisions.

You can sort data by values (smallest to largest and largest to smallest) and members (A to Z and Z to A). If you sort the values of one measure (for example sales volume) from smallest to largest, the other cells in your crosstab are also affected by the sorting. The metadata information therefore remains correct in the crosstab.

The following sorting options are available:

- **Ascending**
If you select ascending sorting, the data is sorted from smallest to largest. Ascending sorting for members results in sorting from A to Z.
- **Descending**
If you select descending sorting, the data is sorted from largest to smallest. Descending sorting for members results in sorting from Z to A.
- **Break Hierarchies**
In the default setting, data is sorted in the hierarchy levels of the crosstab. Parent members are sorted in order, and child members are sorted below each parent member in their own order. If you want to sort data across the entire hierarchy, select the "Break Hierarchies" option.

Related Topics

- [To sort values](#)
- [To sort members](#)

4.6.1 To sort values

1. Select a cell in the crosstab that belongs to the measure you want to use for sorting.
2. Choose **Sort > More Sort Options....**
The "Sort by" dialog box appears.
3. Select the sort options you want to use:
 - **Sort Ascending**
Sorts values from smallest to largest.
 - **Sort Descending**
Sorts values from largest to smallest.
 - **Break Hierarchies**
If you select this option, the values are sorted across hierarchies. Existing hierarchies in the crosstab and their subtotals are not available any more after sorting. In the default setting, the option is not selected.

Note:

If you do not want to change the settings for Break Hierarchies, you can also select the sorting options Sort Ascending and Sort Descending directly in the menu or context menu.

4. Press **OK**.
The sorting is applied to the crosstab.

4.6.2 To sort members

1. Select a cell in the crosstab that belongs to the dimension you want to use for sorting.
2. Choose **Sort > More Sort Options....**
The "Sort by" dialog box appears.
3. Select the sort option you want to use:
 - **Sort Ascending**
Sorts members from A to Z.
 - **Sort Descending**

Sorts members from Z to A.

4. In the **Sort by** list, select the object type for sorting:

- Default (Dimension Members by First Display Type)

The members are sorted by the first display type of the selected dimension.

- Member Display Type

You can select one of the member display types in the list below.

- Measure

If you select this object type, you can select one of the available measures in the list below. With this object type, the sorting is applied to the values of the selected measure and not to a dimension.

Note:

If you do not want to change these settings, you can also select the sorting options Sort Ascending and Sort Descending directly in the menu or context menu.

5. Select **Break Hierarchies** if you want to sort the members across hierarchies. Existing hierarchies in the crosstab and their subtotals are not available any more after sorting.

6. Press **OK**.

The sorting is applied to the crosstab.

4.7 Working with hierarchies

There are two options to work with hierarchical presentations in your analysis: Dimensions with hierarchies and Dimensions displayed as hierarchy.

Including dimensions with hierarchies

Hierarchies for dimensions are created in the data modeling of SAP NetWeaver BW. In SAP NetWeaver BW, they are called "Characteristic Hierarchy". A dimension can contain multiple hierarchies. In a hierarchy, dimension members are organized in a tree structure. For example, the hierarchy of cost centers that are assembled in cost center groups.

Note that dimension with hierarchies can behave differently during filtering and sorting and that there are separate styles for displaying hierarchy levels. The behavior is described in the corresponding chapters.

To find more information about the definition of characteristic hierarchies in SAP NetWeaver BW, go to the SAP Help Portal at "<http://help.sap.com>" **SAP NetWeaver > Select release and language > SAP NetWeaver Library > SAP NetWeaver by Key Capability > Business Intelligence > BI Platform > OLAP > Hierarchies**.

Displaying single dimensions as hierarchy

You can display multiple dimensions in a crosstab as flat presentation or as hierarchy. By default, the flat presentation is displayed. You can change the display to a hierarchical presentation. You can also

use dimensions with hierarchies as a single dimension and combine it with other dimensions to a hierarchical presentation.

Related Topics

- [To include dimensions with hierarchies in an analysis](#)
- [To display single dimensions as hierarchy](#)
- [To filter data by member within hierarchies](#)
- [Sorting data](#)
- [SAP cell styles](#)

4.7.1 To include dimensions with hierarchies in an analysis

By using a data source that contains a dimension with hierarchies, you can define different views on the hierarchies of the dimension. You can also display the members of the dimension in a flat presentation.

As of step two, the steps in the following procedure are optional. You can use the options you need to define your view on the data.

1. Move a dimension with hierarchies to the "Rows" section in the design panel.

Note:

Dimensions can also be moved to the **Columns** section. This procedure describes dimensions in the **Rows** section. Dimensions in columns behave accordingly.

The dimension is displayed in the crosstab. If the dimension contains more than one hierarchy, the displayed hierarchy is selected according to the definition in the BEx Query Designer. Nodes with a + contain subnodes.

2. Select a different hierarchy.
Select a different hierarchy of the dimension in the **<data source>** section in the design panel and move it to the dimension in the **Rows** section.
The data in the crosstab changes according to your selection.
3. Expand/Collapse all nodes.
Select the cell with the dimension name in the crosstab and choose **Hierarchy > Expand to Level > <Level>**. The number of available levels depends from the hierarchy.
The levels in the crosstab are expanded according to your selection. To collapse the levels, choose the first level.
4. Expand/Collapse a single node.
Select a node cell in the crosstab that you want to expand and choose **Hierarchy > Expand** .
The selected node is expanded to the next level. To collapse the subnodes of a selected node, choose **Hierarchy > Collapse**. You can also select the +/- symbols to expand and collapse a single node.
5. Define the available levels of the hierarchy.

Select a member cell of the dimension in the crosstab and choose **Hierarchy > Show Level....**
Select the levels that should be available in the analysis and press **OK**.

Only the selected levels can be displayed in the crosstab. The total is not affected by this selection because the leaf level is always available.

6. Define the sequence of a hierarchy.

By default, a hierarchy is displayed from parent level to child level. To change this presentation, select the node of the first level and choose **Hierarchy > Expand Upwards (Rows)**.

The hierarchy is displayed from child level to parent level.

7. Display the dimension as flat presentation.

Select the **Flat Presentation** node in the <data source> section of the design panel and move it to the dimension in the **Rows** section.

The members of the dimensions are displayed in a flat presentation.

The dimension with hierarchies is displayed according to your definition.

4.7.2 To display single dimensions as hierarchy

If you use multiple dimensions in your analysis, you can display them as flat presentation or as hierarchy in the crosstab.

As of step three, the steps in the following procedure are optional. You can use the options you need to define your view on the data.

1. Move at least two dimensions to the "Rows" section in the design panel.

Note:

Dimensions can also be moved to the **Columns** section. This procedure describes dimension in the **Rows** section. Dimensions in columns behave accordingly.

The dimensions are displayed side by side in the crosstab.

2. Change to hierarchical display.

Select a crosstab cell and choose **Hierarchy > Compact Display in Rows**.

The dimensions in the crosstab are displayed as hierarchy. The first dimension in the **Rows** section is the first level in the hierarchy. Nodes with a **+** contain subnodes.

3. Expand/Collapse all nodes.

Select the cell with the dimension name in the crosstab and choose **Hierarchy > Expand to Dimension > <Dimension Name>**. The number of available dimensions depends from the number of dimensions in the Rows section.

The hierarchy in the crosstab is expanded according to your selection. To collapse the hierarchy, choose the first dimension.

4. Expand/Collapse a single node.

Select a node cell in the crosstab that you want to expand and choose **Hierarchy > Expand Dimension <Dimension Name>**.

The selected node is expanded to the next dimension below. To collapse the subnodes of a selected node, choose **Hierarchy > Collapse**. You can also select the +/- symbols to expand and collapse a single node.

5. Define the sequence of the dimensions in the hierarchy.

The sequence of the dimension in the **Rows** section determines their level in the hierarchy. The first dimension in the design panel is the first level in the hierarchy. By using drag and drop, you can move the dimensions up and down. The levels in the hierarchy are changed immediately.

The hierarchy is displayed from child level to parent level.

6. Display the dimensions again as flat presentation.

Select a crosstab cell and choose **Hierarchy > Compact Display in Rows**.

The dimensions are again displayed side by side in the crosstab.

The dimensions are displayed according to your definition.

4.8 Calculating new measures

Based on measures that are available in your analysis, you can calculate new measures. There are two types of calculations.

For simple calculations, you use two or more available measures as operands and an operator to create a new measure. The new measure is the sum of two available measures, for example.

For dynamic calculations, you only use one available measure as operand and you create a new measure based on this operand. For example, if you use sales volume per region as operand, you can add a new measure that displays the rank of each region according to sales volume. If you now filter out or add new regions to your analysis, the rank numbers are changed dynamically. These calculations are therefore called dynamic calculations.

The newly created measures are added to the crosstab and to the design panel. In the design panel, you can switch the sequence of the measures, edit their name or delete them. Directly on the crosstab, you can also switch the sequence using drag and drop, and edit the name and delete measures with the context menus.

Related Topics

- [To calculate a new measure based on available measures](#)
- [To add a new measure based on one available measure](#)

4.8.1 To calculate a new measure based on available measures

1. Select the measure headings that you want to use.

The first measure that you select is the first operand. Use the **STRG** key to select the next measure that is the second operand. You can also select more than two operands for your calculation.

2. Choose **Calculations > Add Calculation > [Operator]**.

The following operators are available:

- Add
- Subtract
- Multiply
- Divide
- Percentage Difference

The new measure calculates the difference between operand 1 and operand 2 as a percentage.

For example, operand 1 has value 80 and operand 2 has value 20. The new measure displays the value 300%, as the difference is 60 and 60 is 300% of 20.

If operand 2 has a higher value than operand 1, the result is negative. For example, operand 1 has value 20 and operand 2 has value 80. The new measure displays the value -75%, as the difference is -60 and -60 is -75% of 80.

- Percentage Share

The new measure calculates the share of operand 1 in comparison with operand 2 in percentage.

For example, operand 1 has value 20 and operand 2 has value 80. The new measure displays the value 25%, as 20 is 25% of 80.

The new calculated measure is added to the crosstab and the design panel.

4.8.2 To add a new measure based on one available measure

1. Select the measure heading that you want to use.
2. Choose **Calculations > Add Dynamic Calculation > [Operator]**.

The following operators are available:

Option	Description
Moving Minimum Value	<p>The new measure displays the smallest value available up to this point.</p> <p>For example, there are ten rows with values in your crosstab. The new measure starts in the first row with the same value as the original measure. If the second row in the original measure has a higher value than the first, the value of the first row is repeated in the second row of the new measure and so on.</p>
Moving Maximum Value	<p>The new measure displays the highest value available up to this point.</p>
Accumulative Sum	<p>The new measure displays the sum of all values up to this point.</p> <p>For example, the original measure has eight rows with values. In the fourth row of the new measure, the sum of first four rows is displayed.</p>
Accumulative Sum of Rounded Values	<p>The new measure displays the sum of all rounded values up to this point.</p>
Accumulative Count of All Detailed Values	<p>The values per measure are counted and numbered starting with 1 for the first value. If there are 8 rows with values, including zeros, that contribute to the total, the new measure displays the numbers 1 to 8 for the 8 values.</p>
Accumulative Count of All Detailed Values that are Not Zero, Null or Error	<p>The values per measure are counted and numbered starting with 1 for the first value, excluding values that are equal to zero.</p> <p>For values that are equal to zero, the last number is repeated.</p>
Moving Average	<p>The new measure calculates the average of all values up to this point.</p> <p>For example, if there are five rows with values in the crosstab, the new measure calculates in row 2 the average of the values in row one and two, and so on.</p>
Moving Average that is Not Zero, Null or Error	<p>The new measure calculates the average of all values up to this point, excluding values that are equal to zero.</p>

Option	Description
Rank Number	<p>The new measure displays a rank number for each value of the original measure. The highest value in the original measure has the rank number 1.</p> <p>If a value occurs more than once, the values are assigned to the same rank number. If there are two identical values with rank number 4, the next smallest value has rank number 5.</p>
Olympic Rank Number	<p>The Olympic ranked list differs from the basic ranked list as follows: In the Olympic ranked list, when a value occurs more than once, the next smallest value is not assigned the rank incremented by one, but the rank that corresponds to the number of previous values (including the current value).</p> <p>For example, if the rank 4 occurs twice, the new measure displays number 6 for the next smallest value rank.</p>
Percentage Contribution	<p>The new measure calculates the percentage contribution of a value in the original measure to the overall result of the original measure.</p>

The new measure is added to the crosstab and the design panel.

4.9 Defining Conditional Formatting

You can use conditional formatting to highlight important values or unexpected results in your data.

With conditional formatting, you define rules to select different or critical values in a crosstab. Results that fall outside a set of predetermined threshold values are highlighted in color or designated with symbols. This enables you to identify immediately any results that deviate from the expected results. For example, you directly see answers to questions, such as which products have over 10% revenue increase this year or which regions have sales volume higher than a defined amount.

You can add one or more rules to a conditional format and prioritize the application of the rules in your crosstab. You can also define more than one conditional format in a worksheet and decide for each of them when it should be active by toggling it on and off. Existing conditional formats can be edited or deleted.

Note:

The conditional formatting described in this guide is part of the Analysis Add-In. There is also a conditional formatting available in Microsoft Excel.

Related Topics

- [To define a conditional format](#)
- [To edit conditional formats](#)

4.9.1 To define a conditional format

1. Choose **Conditional Formatting > New** .

The "New Conditional Formatting" dialog box appears.

2. In the **Name** field, enter a name for the new conditional format.
3. In the **Based on** list, select the measure you want to apply the conditional format to.
4. In the **Format** list, select the style you want to use for highlighting values.

The following styles are available:

- **Background**: the cells background is colored.
- **Values**: the text of the cells is colored.
- **Status Symbol**: a symbol is displayed in the cells.
- **Trend Ascending**: a colored arrow (green to red) is displayed in the cells.
- **Trend Descending**: a colored arrow (red to green) is displayed in the cells.
- **Trend Grey**: a grey arrow is displayed in the cells.

Every style consists of nine members. The numbers 1 to 9 represent the priority of the rule.

5. Define one or more rules on the **Definition** tab.

- a. Select the priority.

In the first list, select one of the nine priorities of the style selected above. If you define more than one rule in a conditional format, and the rules overlap for values in your crosstab, the priority determines which rule is applied to these cells. You should therefore use a priority only for one rule in a conditional format. Priority one is the highest priority and priority 9 the lowest.

- b. Select an operator.

You can choose one of the following operators: Between, Outside, Greater Than, Greater Than Or Equal To, Less Than, Less Than Or Equal To and Equal To.

- c. Enter a threshold value or value range.

- d. Press **Add**.

The rule is added to the "Rules" area.

6. Optional step: Define the selection for the conditional format.

On the **Selection** tab, you can define for each dimension if the conditional formatting should be applied to "All Members and Totals" or only to the "Members" or "Totals". The default value is All Members and Totals.

7. Optional step: Define the visualization on the **Display** tab.

Under "Apply conditional formatting to:", you can define which parts of the crosstab are affected by conditional formatting: Data Cells, Row Headers and/or Column Headers.

Under "**Apply Visualization to Another Measure:**", you can specify that the conditional format defined for one measure (sales volume for example), is visualized in the data of another measure (sales order items for example). To use this function, select the **Applied to** check box.

8. Press **OK** to apply your rules to the crosstab.

You have created one or more conditional formats. They are saved with the workbook and are available to other users or in other sessions.

4.9.2 To edit conditional formats

Once you have defined conditional formats, you can open and change them, delete them or activate and deactivate them:

1. Choose **Conditional Formatting > [Name of Conditional Formatting]**.

2. Select one of the following options:

- Select **Edit** if you want to change the current definitions of a conditional format. You can add new rules, change existing rules, , or delete existing ones.

To change existing rules, select the **Change** icon in the rule that you want to change, change the definition of the rule and select the **Confirm** icon to change the rule.

To delete existing rules, select the **Delete** icon in the rule that you want to delete.

To apply the changed rule(s) to your data, press **OK**.

- Select **Delete** if you want to delete a conditional format.
- Select or deselect **Active** if you want to toggle the formatting on and off.

4.10 Defining the display of members, measures and totals

When you insert a new crosstab in your analysis, members, measures and totals are displayed according to the default settings. You can change the display of these elements using the toolbar or context menu.

Related Topics

- [To define the members display](#)
- [Defining the measures display](#)
- [Defining the totals display](#)

4.10.1 To define the members display

Members can be displayed as key, text or both. For texts, you can define which text should be displayed.

Key, text and short text of a dimension and its members is set during creation in SAP NetWeaver BW. In the query definition, you can define the member display that should be used as default. In Analysis, you can define the member display for each dimension.

1. Select a member cell of a dimension to define the display for all members of this dimension.
2. Choose **Members** and the display option you want to use. You can select the following display options:
 - **Text:** The members are displayed with their text.
 - **Key:** The members are displayed with their key.
 - **Text and key:** The members are displayed with their text and key.
 - **Key and text:** The members are displayed with their key and text.
3. If you select a display option that contains text, you can choose between two display options for the text. Choose **Members** and one of the following options:
 - **Short Text:** The short text for the members is used for the text display.
 - **Text:** The text for the members is used for the text display.

The members of the selected dimension are displayed in the crosstab according to your definition.

4.10.2 Defining the measures display

To define the measures display, you can specify the number format for each measure in your analysis. This includes the scaling factor and the decimals used.

For all measures in your analysis, you can define in one single step whether the scaling factors should be displayed. If you use currency as unit, you can translate the current currency to the currency you want to use. You can also define how zeros and negative values are displayed in the crosstab.

Related Topics

- [To define the number format](#)
- [To show or hide the scaling factors](#)
- [To define currency translation](#)
- [To define the display of zeros and negative values](#)

4.10.2.1 To define the number format

1. Select a cell from the measure that you want to change.
2. Choose **Measures > Number Format**.
3. In the **Number Format** dialog box, select the **Scaling Factor** and the **Decimal Places** that you want to use for the measure display.
4. Select the **Display Scaling Factor and Units in the Header for All Measures** check box if the scaling factor should be displayed in the crosstab.
5. Press **OK**.

The data cells of the selected measure are displayed according to your definition.

4.10.2.2 To show or hide the scaling factors

1. Select a crosstab cell.
2. Choose **Measures > Show Scaling Factors**.
When the option is active, a check mark is displayed beside the menu item. To deactivate the option, choose it again.

The scaling factors are displayed for all measures according to your definition.

4.10.2.3 To define currency translation

You can translate the currency for all measures of an analysis.

1. Select a crosstab cell.
2. Choose **Measures > Currency Translation**.
3. In the **Currency Conversion** dialog, select a **Target Currency**, a **Currency Conversion Type** and press **OK**.

The currency conversion types are created in SAP NetWeaver BW. For more information, see SAP Help Portal at "<http://help.sap.com>".

Note:

The selection options in the Target Currency and Currency Conversion Type lists are interdependent. Depending on the currency you selected, the system offers you all conversion types that support this currency. To be able to display all conversion types again, choose the standard setting **Display**

Original Currency. Conversely, if you select a particular conversion type, the system shows all currency options for this conversion type.

4. Select the **Use Currency Translation from Query Definition** check box, if the currency translation should be based on the currency definition in the BEx Query Designer.

The currency that you define in the query definition can be different from the currency in the original InfoProvider. For example, in the InfoProvider the currency USD is used and in the query definition, the currency is translated from USD to EUR. If you want to translate the currency in your analysis to GBP, the currency translation is based on EUR if you select this check box and it is based on USD, if you do not select it.

Note:

The check box is only available in the dialog when you select the Advanced Options in the User Settings.

All measures that are scaled in a currency are translated to the selected currency.

Related Topics

- [User settings](#)

4.10.2.4 To define the display of zeros and negative values

1. Select a crosstab cell.
2. Choose **Measures > General Format**.
3. Select an option from the **Display of Negative Values** list.

The following options are available: -X, X-, (X).

4. Select an option from the **Display Zeros as:** list.

The following options are available:

- With Currency/Unit
- Without Currency/Unit
- Empty Cell
- User Defined

If you select this option, you can enter a text that should be displayed in the cells.

5. Press **OK**.

The values are displayed for all measures according to your definition.

You can also suppress zeros if there are rows or columns that only contain zeros.

Related Topics

- [To show/hide zeros in rows and columns](#)

4.10.3 Defining the totals display

For totals, you can define whether they should be displayed in the crosstab. If they are displayed, you can decide where they should be displayed in the crosstab. You can also define how the totals should be calculated.

Related Topics

- [To locate the totals display in the crosstab](#)
- [To show or hide totals in the crosstab](#)
- [To define the calculation of totals](#)

4.10.3.1 To locate the totals display in the crosstab

In the default display, totals are displayed below members and to the right of the members. You have the following options to change the display:

1. Select a crosstab cell.
2. Choose **Totals > Total Rows Above Members**.
The totals are displayed above the members.
3. Choose **Totals > Total Columns Left of Members**.
The totals are displayed left of the members.

The totals are displayed according to your definition. To change the location again, deselect the option.

4.10.3.2 To show or hide totals in the crosstab

1. Select a member cell in the crosstab.
2. To show the totals in the crosstab, choose **Totals > Show Totals**.
3. To hide totals that only have one member, choose **Totals > Hide Totals if Only One Member is Available**.

The totals are shown / hidden according to your definition.

4.10.3.3 To define the calculation of totals

The default setting is to display totals according to the definition in the query. You can calculate with these values to display another result, for example the highest value for an area.

1. Select a measure heading in the crosstab.
2. Choose **Totals > Calculate Totals As > [Option]**.

The following options are available:

Option	Description
Minimum	The smallest value for the corresponding area is displayed as the result.
Maximum	The highest value for the corresponding area is displayed as the result.
Sum	The total of all values for the corresponding area is displayed as the result.
Sum of Rounded Values	If you have set a scaling factor, it is useful to calculate the totals of rounded values locally since the totals can differ considerably (in particular with high scaling factors).
Total Number of Detailed Values	The number of values for the corresponding area is displayed as the result.
Total Number of Detailed Values that are Not Zero, Null or Error	The number of values for the corresponding area, excluding zeros, is displayed as the result.
Standard Deviation	<p>The deviation of the values from its mean is displayed as the result. The more spread out the data is, the higher the deviation. Standard deviation is calculated as the square root of variance.</p> <p>For example, the mean of the following two is the same: 15, 15, 15, 14, 16 and 2, 7, 14, 22, 30. However, the second is clearly more spread out.</p>
Average	The average (total divided by total number of values) of all values for the corresponding area is displayed as the result.
Average of All Detailed Values that are Not Zero, Null or Error	The average (total divided by number) of all values for the corresponding area is displayed as the result, with the exception of zeros, meaning that values equal to zero are not counted.

Option	Description
Median	<p>The system sorts the values internally by size and displays the mean value from the list as the result.</p> <p>For example, if there are five values in the list, the third value is displayed as the result. If the list has an even number of values (for example six), the system adds the two values in the middle of the list together (in this case, the third and fourth values) and divides the result by two. The resulting value is displayed as the result.</p>
Median of Detailed Values that are Not Zero, Null or Error	<p>Excluding the values that are equal to zero, the system sorts the values internally by size and displays the mean value from the list as the result.</p>
Variance	<p>The variance is calculated as the square of the standard deviation and displayed as the result.</p>
Hide	<p>The result is not displayed.</p>
Default	<p>With this setting, the results provided by the query are displayed.</p>

The totals are calculated according to your definition.

4.11 To comment a data cell

With Microsoft Excel functionality, you can add a note to a cell by inserting a comment. You can edit the text in comments and delete comments that you no longer need. You can also choose if the comment text box is shown or hidden in your analysis.

You can add comments to the data cells in the crosstab. The data source must have been refreshed before you add a comment to a data cell. A comment is available internally as long as it is not explicitly deleted. For example, you add a comment to a cell and then change the view on your data. In the new navigation state the cell with the comment is not visible. As soon as you change the navigation state so that the cell with the comment is again visible in the crosstab, the comment is also available.

If you add a comment to a member or header cell in the crosstab, the comment is not available after changing the navigation state.

Note:

If you create a Microsoft Powerpoint presentation with a crosstab that contains comments, the comments will not be available in the presentation.

1. Make sure that the data source has already been refreshed.
2. Select the cell that you want to add a comment.
3. Choose **Insert Comment** in the context menu.
4. In the comment text box, type the comment text.
A small red triangle in the corner of the cell indicates that the cell has a comment attached to it.
5. To edit an existing comment, select the cell and choose **Edit Comment** in the context menu.
6. To keep a comment visible with the cell, select the cell and choose **Show/Hide Comment** in the context menu.
The comment text box is shown permanently in your analysis. You can move the text box using drag and drop.
7. To delete an existing comment, select the cell and choose **Delete Comment** in the context menu.

4.12 To save a workbook

You can save a workbook to the business intelligence platform server or as a local file. The workbook is saved with the current navigation state and selected properties. The workbook properties are part of the **Component** tab in the design panel.

1. In Microsoft Excel 2007, press the Microsoft Office Button and choose **Save Workbook**. If you want to save a workbook locally, press the Microsoft Office Button and choose **Save**.
In Microsoft Excel 2003, choose **Save Workbook** in the Analysis Standard toolbar. If you want to save a workbook locally, choose **File > Save**.
2. Enter your **User**, **Password** and the **WEB Service URL** to the business intelligence platform and press **OK**.
3. Select a folder in the platform folder structure where you want to save the workbook.
4. Enter a name for the workbook.
The current workbook name is entered as default. You can change this name.
5. Press **Save**.
The workbook is saved to the business intelligence platform server.

The workbook is saved. You and other user with access to the server can open and analyze it.

Related Topics

- [Working with Analysis in Microsoft Excel 2007](#)
- [Working with Analysis in Microsoft Excel 2003](#)
- [The Components tab](#)

Analyzing SAP HANA data

5.1 Analyzing SAP HANA data sources

After configuring a connection to SAP HANA, you can analyze information models developed with SAP HANA as data sources in Analysis, for example an analytic view. With SAP HANA data sources, you can use the following Analysis functions to analyze your data:

- Inserting a crosstab with SAP HANA data
You can also insert more than one data source in a workbook.
- Saving/Opening a workbook to/from the business intelligence platform server
- Undo/Redo
- Defining style sets for crosstabs
- Swapping axes
- Analyzing data with the design panel
- Filtering members
- Sorting data by members and by measures
- Inserting charts
- Inserting filter components
- Converting crosstab cells to formula
- Creating Presentations
- Pause Refresh

Planning Data

In Analysis, you can plan business data based on the current data in your data source. You can enter the planning data manually or you can enter planning data automatically using planning functions and planning sequences of SAP NetWeaver BW Integrated Planning. For planning data, you have to use a BW query or query view as data source that is defined as input-ready in BEx Query Designer.

When you work with input-ready data sources, Analysis offers two different modes: display mode and change mode. You use change mode to enter planning data in an input-ready query. If you want to unlock a data source for another user or avoid changing data accidentally, you can switch to display mode. You can also undo your changes and go back to the previous saved state or to the previous consistent client state. You can also lock and unlock selected crosstab cells for changes.

After entering the planning data, you can recalculate or save it. When you recalculate the planning data, all changed values are transferred to the planning buffer and checked for consistency. If the changed data entries are correct, the recalculated data is available and visible in all other components of the workbook. Recalculated data is not saved to the InfoProvider though. If you save the planning data, all transferred and changed values are saved to the InfoProvider.

In Microsoft Excel 2003, the planning functionality is available in a separate toolbar. In Microsoft Excel 2007, the planning functionality is available in a separate group in the ribbon. You can define in the user settings if the planning group should be displayed in the ribbon.

Related Topics

- [To plan data manually](#)
- [To plan data with planning objects](#)
- [To undo changes while planning data](#)
- [To lock cells](#)
- [To recalculate planning data](#)
- [To save planning data](#)
- [To switch between display and change mode](#)
- [User settings](#)

6.1 To plan data manually

You can enter the planning data manually in a crosstab in your analysis. To enter planning data, you can either change the values in existing data cells in the crosstab or you can add new lines to crosstabs in change mode.

1. Enter an input-ready data source or open a workbook with an input-ready data source.

The crosstab is visible in change mode. In change mode, read-only cells of the data source are displayed with different background colors. If you switch to display mode, the distinct visualization of input-ready and read-only cells is not visible.

Note:

Microsoft Excel cells are always input enabled. That means you can also enter values in the read-only cells of the data source. These values will not affect your planning though. If the read-only cells are displayed with a different background color, you can avoid entering data in read-only cells.

If a data source is not displayed in change mode, you can check the following settings:

- Check on the **Components** tab in the design panel if the correct planning system is connected.

In the default setting, the planning system is the first system you connect to after opening a workbook. If a workbook contains objects from different systems, you can select one of these systems as the planning system in the list. The changed planning system will only be effective after closing and reopening the workbook.

- Check on the **Components** tab in the design panel if the check box **Planning: Open Data Source Input-Enabled** is selected.

If the check box is selected, the input-ready data source is opened in change mode. If it is not selected, the data source is opened in display mode.

- If a data source is locked by another user, a message is displayed after opening.

You can also use the property `IsDataSourceEditable` of the API method `SAPGetProperty` to check if a data source is input-ready.

2. You can enter planning data by changing data in existing data cells of the crosstab.

The format of an entered value is immediately validated. If the format is not correct, a message is displayed.

To check the consistency of the new values and apply it to the crosstab, choose **Recalculate**.

3. You can enter planning data in new lines added to the crosstab.

You can define the number of new lines on the **Components** tab in the design panel. The default number of lines is 5. New lines for planning are added to the bottom of the crosstab. In display mode, the new lines are not visible.

To enter data in the data cells, you have to enter the complete member combination for the data cell. For example, the dimensions Country, Region and City are in the rows of your crosstab. If you want to enter a planning value for Paris, you also have to enter the region and country member in the corresponding line. You can enter the member manually or select values with input help. The entered values are immediately validated for format and master data. If a value is not correct, a message is displayed.

To check the consistency of the new values and apply them to the crosstab, choose **Recalculate**.

4. You can undo all changes that are not recalculated or undo recalculated changes that are not saved.

Select **Back > To Previous Consistent Client State / Saved State**.

5. Choose **Save** to save the planning data to the system.

The changed data is saved in the data source.

Related Topics

- [To recalculate planning data](#)
- [To save planning data](#)
- [To undo changes while planning data](#)
- [To switch between display and change mode](#)
- [The Components tab](#)
- [SAP cell styles](#)

6.2 To plan data with planning objects

You can use two planning objects: planning functions and planning sequences. Planning functions allow system-based processing or generation of data. A planning sequence is a sequence of planning functions. Both objects are created with Planning Modeler in SAP NetWeaver BW.

The following procedure explains the steps to be done in **Analysis** when working with planning objects. For more information about SAP NetWeaver BW Integrated Planning, Planning Modeler and objects like planning function or aggregation level, see the corresponding documentation on the SAP Help Portal.

1. Insert an input-ready data source or open a workbook with an input-ready data source.

Note:

This step is optional. You can also use planning objects without visualization in a crosstab.

2. Add a planning object to the workbook.

On the **Components** tab in the design panel, choose **Use Planning Function** or **Use Planning Sequence** in the context menu of the workbook. In the **Open** dialog box, search for the planning object, select an object and press **OK**.

The node Planning Objects is added to the workbook on the **Components** tab. All planning objects are assigned to this node.

3. Check on the **Components** tab in the design panel if the correct planning system is connected. In the default setting, the planning system is the first system you connect to after opening a workbook. If a workbook contains objects from different systems, you can select one of these systems as the planning system in the list. The changed planning system will only be effective after closing and reopening the workbook.
4. Select the **Process Changed Data** check box if you want to use process changed data. If you select this check box, you can define that only data changed by the user in the current session since the last save will be processed. You can specify which aggregation level should be used as a filter to define the changed data. In the default setting, the aggregation level the planning function is built on is selected for a planning function. As a planning sequence consists of several planning functions, no aggregation level is selected automatically. You can select an aggregation level for a planning sequence manually.

5. Define the variables values of the planning object.

On the **Variables of Planning Sequence / Functions** tab, select the source and define the values:

- For source **Member**, you can select the members with input help.
- For source **Cell**, you can define the cell that contains the values.

Variables act as placeholders for members, hierarchies and numbers. They can be defined in BEx Query Designer or in Planning Modeler. All variable values are valid locally in the planning object. To plan data for two different years for example, you can insert the same planning function twice in a workbook and enter different values for the dimension year.

6. Define the filter for the planning function.

On the **Filter** tab, you get a list of all dimensions of the aggregation level.

a. Define the filter values for the dimensions as required.

For source **Member**, you can select the members with input help.

For source **Cell**, you can define the cell that contains the values.

For source **Link**, the current filter values in the data source are used.

Note:

For dimension with source Link, you can also use the API method SAPSetFilter, to define the filter values.

b. You can also select the filter the planning function is linked to:

- Press the **Open Filter** button to insert a filter object.

The filter object can be created in the BEx Query Designer and in the Planning Modeler.

- Press the list button to select the data source you want to use for planning. The data source should already be used in the workbook.

When you insert a filter object or data source, the source type changes from Member to Link for all dimensions that have not previously been defined, and the current filter values in the data source or filter object are used.

c. Redefine filter values.

You can change the filter definition for a planning function at any time. After inserting a data source, for example, you can change the source from Link to Cell and define a cell that contains the values that should be used as filter.

7. Choose **Execute Planning Function** in the context menu of a planning function on the **Components** tab to execute the object.

When working with planning sequences, choose **Execute Planning Sequence** in the context menu of a planning sequence on the **Components** tab to execute the object.

Note:

The changed data is recalculated automatically when you execute a planning object. To undo all changes since the last save, choose **Back > To Previous Saved State**.

8. Optional step: Delete a planning object from the workbook.

Choose **Delete** in the context menu of the planning object on the **Components** tab.

9. Choose **Save** to save the planning data to the system.

The changed data is saved in the data source.

6.3 To recalculate planning data

1. Enter or change planning data.
2. Choose **Recalculate**.

All changed values are checked for consistency. If the changed data entries are correct, the recalculated data is available, the totals are updated and the new values are visible in all other components of the workbook. The recalculation is executed for all data sources in the workbook that are available in the selected planning system.

You can also use the PlanDataTransfer command from API method SAPExecuteCommand to recalculate planning data.

Related Topics

- [SAPExecuteCommand](#)

6.4 To save planning data

1. Enter or change planning data.
2. Choose **Save**.

All changed values are checked for consistency. If the changed data entries are correct, the data is saved in the data sources and a message is displayed that the data was saved. The saving is executed for all data sources in the workbook that are available in the selected planning system.

You can also use the PlanDataSave command from API method SAPExecuteCommand to save planning data.

Related Topics

- [SAPExecuteCommand](#)

6.5 To lock cells

You can lock input-ready crosstab and total cells in Analysis. A locked cell can not be changed manually when the crosstab is recalculated. If you execute a planning function or sequence the locked cells are unlocked automatically.

Note:

Microsoft Excel cells are always input enabled. That means you can also enter values in the locked cells of the data source. These values will not affect your planning though.

The cells are locked for the current user session only. You can also unlock the cells during the session.

1. Select the crosstab cells you want to lock.

The selected cells have to be input enabled and have not been changed since the last recalculation.

If you select a mixture of input-ready and already locked cells, all selected cells will be locked.

2. Choose **Lock Cells** in the planning group.

You can also use the context menu to lock and unlock the cells.

The selected cell are locked for changes. The locked cells are displayed in a different cell style.

To unlock the cells, select all cells you want to unlock and choose **Unlock Cells**.

6.6 To undo changes while planning data

1. Enter or change planning data.
2. To undo your changes, you have the following options:

- Choose **Back > To Previous Saved State**.

With this option, you undo all changes since the last save of your data. The undo is executed for all data sources in the workbook that are available in the selected planning system.

You can also use the PlanDataReset command from API method SAPExecuteCommand to revert to the last saved state of your planning data.

- Choose **Back > To Previous Consistent Client State**.

With this option, you undo all changes since the last successful recalculation of your data. The undo is executed for all data sources in the workbook that are available in the selected planning system.

You can also use the PlanDataClientReset command from API method SAPExecuteCommand to revert to the last recalculated state of your planning data.

Related Topics

- [SAPExecuteCommand](#)

6.7 To switch between display and change mode

When you work with input-ready data sources, Analysis offers two different modes: display mode and change mode. You use change mode to enter planning data and to recalculate or save the changed data to the BW system. If a user is working with an input-ready data source in change mode, this data source is locked for the other users. If the user switches to display mode, another user can work with this data source in change mode.

When you open a workbook containing input-ready data sources, the crosstabs of these data sources are displayed in change mode. In change mode, new lines for planning are added to the bottom of the crosstab, and read-only cells of the data source are displayed with different background colors. The Add-In includes SAP standard styles for the visualization (SAPEditableDataCell, SAPEditableDataTotalCell, SAPReadonlyDataCell and SAPReadonlyDataTotalCell). You can change the cell styles as required. In display mode, the distinct visualization of input-ready and read-only cells is not visible.

Switch to display mode

You work in a workbook that includes input-ready data sources.

1. Enter or change planning data.
2. Choose **Display** to switch to display mode.
3. A dialog box asks if you want to save your changes before switching to display mode.

Press **Yes** to save your changes or **No** to discard them.

The data sources are switched to display mode. The distinct visualization of input-ready and read-only cells is not visible any more and the new lines at the bottom of the crosstab are deleted. The data sources can now be used in change mode by another user.

You can also use the `PlanDataToDisplayMode` command from API method `SAPExecuteCommand` to switch to display mode.

Switch to change mode

You work in a workbook that includes input-ready data sources. You are currently working in display mode.

1. Check on the **Components** tab in the design panel that the correct planning system is connected.
2. Check on the **Components** tab in the design panel that the check box **Planning: Open Data Source Input-Enabled** is selected for the data sources you want to change.
3. Choose **Change** to switch to change mode.

All data sources in the workbook that are available in the selected planning system and are selected as input-enabled are switched to change mode. The distinct visualization of input-ready and read-only cells is applied, and the new lines are added at the bottom of the corresponding crosstabs.

Note:

If a data source is locked by another user, a message is displayed.

You can also use the `PlanDataToChangeMode` command from API method `SAPExecuteCommand` to switch to change mode.

Related Topics

- [The Components tab](#)

- [SAP cell styles](#)

Creating Presentations

7.1 To create a slide out of Microsoft Excel

You have inserted a data source in Analysis, edition for Microsoft Excel.

Out of the edition for Microsoft Excel, you can create a slide with a crosstab or chart that is currently displayed in the workbook. In Microsoft PowerPoint, the current navigation state of the objects in Excel is displayed, but the objects exist independently in Excel and PowerPoint. After the creation of the slide, navigation steps in Excel do not affect the objects in PowerPoint and vice versa.

1. In the edition for Microsoft Excel, select the crosstab or chart that should be displayed on the slide.
2. Choose **Create Slide**.

The edition for PowerPoint opens automatically.

3. If you are not already logged on to the system in the edition for Microsoft PowerPoint, enter your **User** and **Password** and press **OK**.

If a crosstab includes more rows than can be displayed on a slide, the **Fit Table** dialog box appears. In this dialog, you can define if the crosstab should be splitted to multiple slides or displayed abbreviated in one slide.

4. Optional step: Change the view on the data.

You can change the view on the data in Microsoft PowerPoint using the functions of the Analysis ribbon.

You can use the undo/redo in the Analysis ribbon to undo and redo steps that you executed with functions in the Analysis ribbon. To undo/redo steps that you executed with Microsoft PowerPoint, for example the deletion of a crosstab, you should use the undo/redo of Microsoft PowerPoint.

5. Optional step: Change the formatting of a crosstab.

To change the formatting of a crosstab, select the crosstab and choose **Table Tools > Design > Table styles**. Select a style to apply it to the crosstab.

In the **Table Style Options** group, you can define the display of rows and columns. If you select **Header Row** and **First Column**, the formatting is only applied to the first header row and first column even if your Analysis crosstab contains more than one header rows or columns with member information. The formatting will be applied to these rows and columns after the next Analysis refresh.

Note:

You should not use the other options in the Table Tools ribbon to format an Analysis crosstab. If you use them, your changes will be overwritten with the next Analysis refresh.

6. Optional step: You can define specific settings for Analysis in Microsoft Powerpoint in the **Presentation Settings** dialog.

To open this dialog, choose **Settings > Presentation Settings**.

The selected object is displayed on a Microsoft PowerPoint slide.

Corresponding to workbooks, you can save a presentation to the business intelligence platform server and open it from there. Choose the Microsoft Office Button and **Save/Open Presentation**.

Related Topics

- [Working with Analysis in Microsoft PowerPoint 2007](#)
- [To open a workbook](#)
- [To save a workbook](#)
- [Presentation settings](#)

7.2 Presentation settings

In the **Presentation Settings** dialog, you can define specific settings for Analysis in Microsoft Powerpoint. In Microsoft Excel, these settings are available on the **Components** tab in the design panel.

To open this dialog in Microsoft PowerPoint, choose **Settings > Presentation Settings**.

Refresh Presentation on Opening

If you select this check box, the data sources in the presentation are refreshed every time the presentation is opened.

If this check box is not selected, the data in the presentation is not automatically refreshed on opening. You can refresh the data sources manually with choosing **Refresh All** in the menu.

Force Prompt for Initial Refresh

If you select this check box, the prompting dialog is displayed on every refresh.

Store Prompts with Presentation

If you select this check box, the defined prompt values are saved with the presentation.

Remove Data Before Saving

If you select this check box, the presentation is saved without the data. When you reopen the presentation, no data is displayed. To display the data, refresh the data sources manually with choosing **Refresh All** in the menu.

Using Analysis in other SAP BusinessObjects Applications

After performing an analysis, you might want to use other SAP BusinessObjects applications to communicate your findings and share the analysis with others.

To use an analysis in other applications, you save it as an analysis view. An analysis view is a saved navigation state of an analysis including applied filters, hierarchies, and available metadata.

Analysis views can be saved with SAP BusinessObjects Analysis, edition for Microsoft Office, and with SAP BusinessObjects Analysis, edition for OLAP. They can be inserted in both editions of SAP BusinessObjects Analysis.

Like workbooks, analysis views are saved to the business intelligence platform server. While a workbook can comprise several analyses, an analysis view comprises just one.

Related Topics

- [To save an analysis view](#)
- [To insert an analysis view](#)

8.1 To save an analysis view

You have to be connected to a business intelligence platform server.

1. In the worksheet, select the crosstab component that you want to save as an analysis view.
2. Choose **Save Analysis View** in the context menu.

Note:

If you are not connected to a platform server, the entry is not available in the context menu.

3. Select the location to save your analysis view to.
4. Type a filename for the analysis view and press **Save**.

The analysis view is saved to the repository.

8.2 To insert an analysis view

1. In your worksheet, select the cell where you want to insert the analysis view.
2. Choose **Insert > Insert Analysis View**.

If you are not already connected to a business intelligence platform server, connect to the server where the analysis view you want to open is saved.

3. In the folder list, select an analysis view and press **OK**.

The selected analysis view is added to the worksheet. You can change the displayed data set and create new views on the data as required.

Settings

In Analysis, edition for Microsoft Excel, you can change user settings in according with your personal preferences. The "User Settings" dialog box is divided into two tab pages, each providing different settings: **User Settings** and **Support Settings**.

To open the dialog, choose **Analysis > Settings**.

Related Topics

- [User settings](#)
- [Support settings](#)

9.1 User settings

On the **User Settings** tab, you can define the following settings.

In Analysis for Microsoft PowerPoint, additional settings are available in the **Presentation Settings** dialog.

Recently Used List

When you choose **Insert** in the menu to insert a new data source in the workbook, you get a list of recently used data source for quick selection. To define the number of listed data sources, enter the required number in the **Maximum Number of Entries in Recent list** field.

If you press the **Delete Recently Used List** button, the history of recently used data sources is deleted.

Number of Members Displayed in Filter Dialog

Here, you can enter the maximum number of members displayed in the **Filter by Member** dialog for selection. To define the number of members displayed, enter the required number in the **Maximum Number of Members Displayed in Filter Dialog** field. If you filter on a dimension that contains more members than defined in the user settings, a message is displayed in the filter dialog.

Note:

You can also define settings at workbook level. These workbook properties are part of the **Component** tab in the design panel.

Default Workbook Location

You can define the the location of your default workbook. The default workbook is used when you open a data source by choosing **Open > Data Source**. The data source will be opened with the styles, formatting and the workbook settings of the default workbook.

Enable Analysis Office if Microsoft Office is Started from Another Program

Select this check box, if the Analysis Add-In should always be opened and available in the ribbon when you open an Analysis workbook or presentation from another program without opening Microsoft Excel or Powerpoint with the Add-In before.

To use this setting, you must have enabled the Add-In in Microsoft Excel / Powerpoint.

Advanced Options

Select this check box if you want the check box **Use Currency Translation from Query Definition** to be available in the currency translation dialog for measures.

Show Planning toolbar

Select this check box if you want the planning group to be displayed in the ribbon.

Low Speed Connection (Reduced Network Traffic)

This setting is recommended when working in a WAN environment. By default, this setting is not selected.

Related Topics

- [The Components tab](#)
- [Using a default workbook to open a data source](#)
- [Presentation settings](#)

9.2 Support settings

You can record traces to analyze problems in Analysis workbooks. A trace is a self-contained quantity of logged user actions. On the **Support Settings** tab, you can make various settings for the trace level and the location of the log file. The log file with the recorded traces helps to resolve problems that you report to SAP as customer messages.

Folder for Log File

To set the location of the log file in your file system, press **Explore**. The default log file location is `C:\Documents and Settings\\Local Settings\Temp\SapAdvancedAnalysis.log`.

The log files for both Microsoft Excel files and Microsoft PowerPoint files of Analysis are stored in this folder. The names of the log files are:

- *SAPAdvancedAnalysisXLS.log* for Microsoft Excel log files of Analysis
- *SAPAdvancedAnalysisPPT.log* for Microsoft PowerPoint log files of Analysis

The system writes the trace information to the log file until the file reaches the maximum size of 2048 KB. The log file can be PC-specific (per installation) or user-specific, depending on the folder path in the file system (\MyDocuments\SAPAdvancedAnalysis for example). If the log file exceeds the maximum size, the system creates a copy and saves it with a timestamp. The copied log files are stored in the same folder.

Log Severity

This setting defines the amount of information stored in the log file. The following options are available:

- **Information** : The system stores exceptions, error messages, warnings and information messages ("The workbook was saved." for example) in the log file. This is the default option.
- **Warning** : The system stores exceptions, error messages and warnings in the log file.
- **Error** : The system stores exceptions and error messages in the log file.
- **Critical** : The system stores exceptions in the log file.
- **Debugging Information** : The system stores exceptions, error messages, warnings, information messages and debugging information in the log file.

Enable BW Server Tracing

To activate the SAP NetWeaver BW trace tool environment, select the **Enable BW Server Tracing** check box.

The trace tool environment (transaction code RSTT in the connected BW system) has special tools to log and playback traces and process automatic regression tests.

For more information about the trace tool environment, see SAP Help Portal at "<http://help.sap.com>" **SAP NetWeaver > SAP NetWeaver by Key Capability > Information Integration by Key Capability > Business Intelligence > BI Platform > OLAP > Performance Optimization > BEx Monitor > Trace Tool Environment.**

Enable Profiling

To activate the query runtime statistics of SAP NetWeaver BW, select the **Enable Profiling** check box.

Using the query runtime statistics, you can find out how much time the execution of certain user actions require in Analysis and in the analytic engine of BW. The system records the performance-critical parts of the processing (statistics events). It calculates the net times by calculating the runtime of an event using the difference between the start and end times (minus the times for other events called from within the event).

You can check the statistics in the **Statistics...** entry in the Analysis help menu. The **Statistics...** entry is only visible if statistics are available.

For more information about the query runtime statistics, see SAP Help Portal at "<http://help.sap.com>" **SAP NetWeaver > SAP NetWeaver by Key Capability > Information Integration by Key Capability > Business Intelligence > BI Platform > OLAP > Performance Optimization > BEx Monitor > BI Query Runtime Statistics.**

Display Technical Names

To switch on the technical names of the dimensions in your Analysis workbook, select the **Display Technical Names** check box. The dimensions are displayed with technical name and text.

Enable NCO Tracing

This setting may be used for SAP error handling.

Troubleshooting

10.1 To enable the Analysis Add-In after system crash (Microsoft Office 2007 and 2010)

If Microsoft Excel 2007/2010 or Microsoft PowerPoint 2007/2010 crashes, and you have to close the application, the Analysis Add-In might be disabled by the Microsoft application. If this happens, you have to re-enable the Analysis Add-In in Microsoft Excel or Microsoft PowerPoint.

1. Open Analysis for Microsoft Excel or Microsoft PowerPoint.
After the system crash the Analysis Add-In is not visible in the menu.
2. Press the **Microsoft Office Button** in Microsoft Office 2007.
Choose File in Microsoft Office 2010.
3. In Microsoft Excel, press **Excel Options**. In Microsoft PowerPoint, press **PowerPoint Options**.
4. In the "Excel Options" dialog box and in the "PowerPoint Options" dialog box in the categories pane, select **Add-Ins**.
5. In the **Manage** box, select **Disabled Items**.
6. Press **Go...**
7. In the "Disabled Items" dialog box, select the Analysis Add-In.
8. Press **Enable**.
9. In the **Manage** box, select **COM Add-Ins**.
10. Press **Go...**
11. In the "COM Add-Ins" dialog box, make sure that **Analysis** option is activated.
12. Press **OK**.

The **Analysis** ribbon is available again.

10.2 To enable the Analysis Add-In after system crash (Microsoft Excel 2003)

If Microsoft Excel 2003 crashes, and you have to close the application, the entries in the **Analysis** menu might be disabled by the Microsoft application. If this happens, you have to re-enable the Analysis Add-In in Microsoft Excel.

1. Open Analysis for Microsoft Excel.

2. In the **Tools** menu, choose **Add-Ins....**
3. In the "Add-Ins" dialog box, select the options **Analysis ToolPak** and **Analysis ToolPak - VBA**.
4. Press **OK**.
5. Close Microsoft Excel and open Analysis for Microsoft Excel again.

The entries in the **Analysis** menu are available again.

10.3 Solving issues regarding the creation of Microsoft PowerPoint slides

If users encounter problems when executing the function **Create Slide** in Analysis for Microsoft Excel, we recommend that you do the following:

Make sure that users start Analysis for Microsoft PowerPoint manually before they execute the function **Create Slide** in Analysis for Microsoft Excel.

Users can start Analysis for Microsoft PowerPoint in the Windows directory under **Start > Programs > SAP Business Objects > Analysis for Microsoft PowerPoint** or by double-clicking the corresponding desktop icon.

More Information

Information Resource	Location
SAP BusinessObjects product information	http://www.sap.com
SAP Help Portal	<p>Navigate to http://help.sap.com/businessobjects and on the "SAP BusinessObjects Overview" side panel click All Products.</p> <p>You can access the most up-to-date documentation covering all SAP BusinessObjects products and their deployment at the SAP Help Portal. You can download PDF versions or installable HTML libraries.</p> <p>Certain guides are stored on the SAP Service Marketplace and are not available from the SAP Help Portal. These guides are listed on the Help Portal accompanied by a link to the SAP Service Marketplace. Customers with a maintenance agreement have an authorized user ID to access this site. To obtain an ID, contact your customer support representative.</p>
SAP Service Marketplace	<p>http://service.sap.com/bosap-support > Documentation</p> <ul style="list-style-type: none"> • Installation guides: https://service.sap.com/bosap-instguides • Release notes: http://service.sap.com/releasenotes <p>The SAP Service Marketplace stores certain installation guides, upgrade and migration guides, deployment guides, release notes and Supported Platforms documents. Customers with a maintenance agreement have an authorized user ID to access this site. Contact your customer support representative to obtain an ID. If you are redirected to the SAP Service Marketplace from the SAP Help Portal, use the menu in the navigation pane on the left to locate the category containing the documentation you want to access.</p>
Docupedia	<p>https://cw.sdn.sap.com/cw/community/docupedia</p> <p>Docupedia provides additional documentation resources, a collaborative authoring environment, and an interactive feedback channel.</p>
Developer resources	<p>https://boc.sdn.sap.com/</p> <p>https://www.sdn.sap.com/irj/sdn/businessobjects-sdklibrary</p>

Information Resource	Location
SAP BusinessObjects articles on the SAP Community Network	https://www.sdn.sap.com/irj/boc/businessobjects-articles These articles were formerly known as technical papers.
Notes	https://service.sap.com/notes These notes were formerly known as Knowledge Base articles.
Forums on the SAP Community Network	https://www.sdn.sap.com/irj/scn/forums
Training	http://www.sap.com/services/education From traditional classroom learning to targeted e-learning seminars, we can offer a training package to suit your learning needs and preferred learning style.
Online customer support	http://service.sap.com/bosap-support The SAP Support Portal contains information about Customer Support programs and services. It also has links to a wide range of technical information and downloads. Customers with a maintenance agreement have an authorized user ID to access this site. To obtain an ID, contact your customer support representative.
Consulting	http://www.sap.com/services/bysubject/businessobjectsconsulting Consultants can accompany you from the initial analysis stage to the delivery of your deployment project. Expertise is available in topics such as relational and multidimensional databases, connectivity, database design tools, and customized embedding technology.