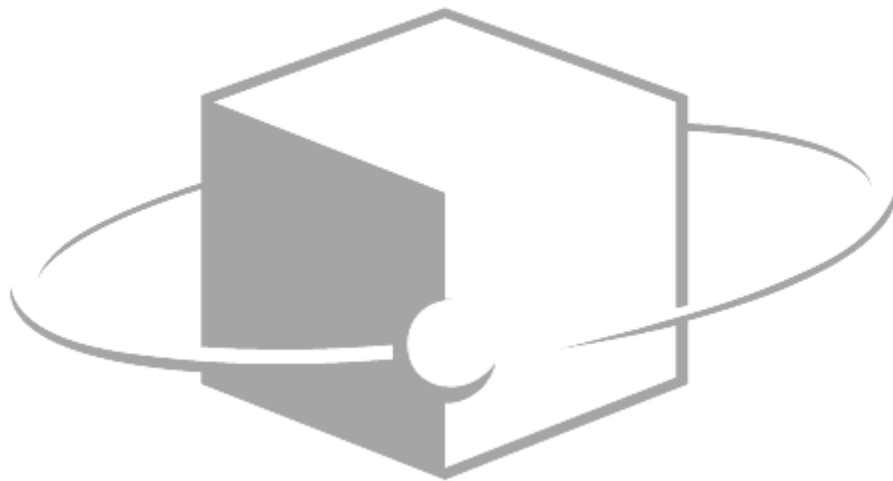

User Manual

instantOLAP

Version 2.6.1
17.11.2009



Copyright

Copyright (C) 2002-2008 Thomas Behrends Softwareentwicklung e.K. All rights reserved.

This manual, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. The content of this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Thomas Behrends. Thomas Behrends assumes no responsibility or liability for any errors or inaccuracies that may appear in this book.

Except as permitted by such license, no part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Thomas Behrends.

instantOLAP is a registered trademark of Thomas Behrends. Windows is a registered trademark of Microsoft Corporation. All other product and company names are trademarks or registered trademarks of their respective holders.

Contents

Copyright	3
Contents	4
Typographical Conventions	7
CHAPTER 1: Using the Web-Portal	9
Starting the Web-Frontend	10
Opening the Portal in your Browser	10
Logging in	10
Logging out	11
The portal structure	12
Portal layout	12
Navigation and explorers	12
The repository explorer	12
The search explorer	14
The bookmark explorer	15
The model explorer	16
The tool explorer	17
The navigator	17
The window manager menu	19
Error messages and other notifications	19
Using queries	20
Opening queries	20
The toolbar	20
The selector bar	21
Executing the query	23
Working with the query	23
Exporting queries	24
Creating bookmarks	25
Creating snapshots	26
Subscribing queries	27
Pivot-tables	28
Opening existing pivot tables	28
Creating a new pivot table	28
The analyzer	29
Working with pivot tables	31
Editing pivot tables	32
Charts	36
The expression editor	37
Working with documents	44
Opening documents	44
Creating new HTML pages	44
Editing HTML pages	44
Uploading documents	45
Tools	46
The Automation manager	46
Navigator	49
Change password	49

CHAPTER 2: Using the Workbench	51
System requirements	52
Workbench installation	53
Installation using the Windows-Installer	53
Installation using Java Webstart	53
Logging in	55
Editing the server list	55
Logging in	56
The desktop	57
The Workbench menu	57
The Toolbar	58
The Explorer	59
Working with the repository	60
Working with files	60
Working with folders	62
Import and export of files and folders	65
Manging folders	67
The Info-Editor	67
The Access-Editor	69
Tools	72
User-Manager	72
Automation-Manager	76
Model-Manager	80
Process-Manager	83
Session-Manager	84
License-Manager	85
The Model-Query tool	88
The Logging-Panel	89
 CHAPTER 3: Using the Query-Editor	 93
Starting the editor	94
Starting the editor in the Workbench	94
Structure of the editor	97
The worksheet	97
The Model-Explorer	98
The Property-Editor	99
The toolbar	102
The preview panel	102
Editing queries	104
Editing the worksheet	105
Editing selectors	107
Editing outer blocks	114
Editing inner blocks	116
Editing pivot-tables	120
Editing comments	130
 CHAPTER 4: Using the Config-Editor	 133
Starting the Config-Editor	134
Creating new configurations	134
Opening existing configurations	135
Structure of the Config-Editor	136

The Worksheet	136
The Configuration explorer	137
The Property-Editor	138
The toolbar	141
The SQL-query tool	141
Editing configurations	143
Editing datasources	143
Editing dimensions	153
Limiting the access to dimensions	157
Editing Cubes and Stores	158
Editing Formulas	166
Editing caches	168
Importing other models	169
Basic configuration settings	169
CHAPTER 5: Concepts	171
A quick overview	172
Models	173
Models	173
Dimensions, Keys and Hierarchies	173
Facts	175
Cubes	176
Structure of a query	177
Selectors	177
Outer blocks	181
Inner blocks	182
Pivot-tables	184
Comments	188
Filters and subcubes	189
How instantOLAP works	191
Configurations	191
Dimensions	192
Facts	196
Cubes	197
Formulas	201
Caches	202
Importing other models	203
Good model design	205
Make models reusable	205
What should become a dimension, attribute or fact?	205
Naming conventions	206

Typographical Conventions

Before you start using this guide, it is important to understand the terms and typographical conventions used in the documentation. The following kinds of formatting in the text identify special information.

Formatting convention	Type of Information
Special Bold	Items you must select, such as menu options, command buttons, or items in a list.
Brackets (<>)	Used for variable expressions such as parameters.
Monospace font	Marks code examples or an expression-syntax.
<i>Emphasis</i>	Use to emphasize the importance of a point.
CAPITALS	Names of keys on the keyboard. for example, SHIFT, CTRL, or ALT.
KEY+KEY	Key combinations for which the user must press and hold down one key and then press another, for example, CTRL+P, or ALT+F4.

CHAPTER 1:

Using the Web-Portal

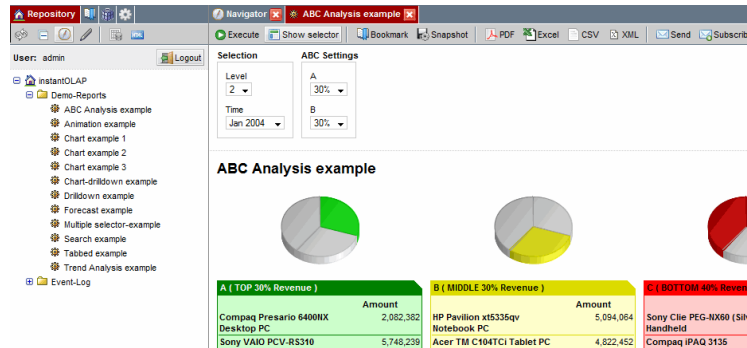
Contents of this chapter:

Starting the Web-Frontend	10
The portal structure	12
Using queries	20
Pivot-tables	28
Working with documents	44
Tools	46

Starting the Web-Frontend

Opening the Portal in your Browser

The instantOLAP
Web-Frontend



Open a browser and enter the location (URL) of your instantOLAP server. The URL of an instantOLAP-Server is `http://<servername>[:<port>]/iolap` or `https://<servername>[:<port>]/iolap`, where `<servername>` must be replaced by the name or ip-address of the server and `<port>` must be replaced by the port of the server. If the port is 80, you can leave it out (this is the standard-port for web-applications). If your server is available only under a secure https-connection, you must use the protocol "https" at the beginning of the URL.

E.g. an URL `http://localhost:8080/iolap` connects to a local instantOLAP server running on your machine on port 8080. An URL `https://10.0.0.10/iolap` connects to a server running on server with the IP-address 10.0.0.10 and the default port. Depending on your installation you will be allowed to use the application as a guest. If working as a guest is allowed, you will immediately see the OLAP-frontend and the public reports. Otherwise, you will be directed to the login-dialog and have to log in first.

Logging in

The login button



To log into the server with your own account, just locate the server or use the Login button at the upper left of the Browser if your server starts in the guest mode. After pressing the Login button, the login dialog will appear.

The login-dialog

The login dialog box contains the following text and fields:

Please type in your user-name and password and press OK to login to the system.

User:

Password:

Remember me:

Server: localhost / instantOLAP 2.5.0 build 2255

Optionally you can use the "Remember me" checkbox to automatically login with your account the next time you open the portal within the next 30 days. You need to enable cookies in your browser to use this feature.

If your input was correct you will be logged in. Now you can see all the folders, reports and snapshots in the navigation-sidebar to which you have access to. If an error occurred (e.g. when you mistyped your password), the error message will be displayed at the bottom of the dialog. The following errors may occur when logging in:

- **Wrong username or password:** You mistyped your username or password and have to enter the data again.
- **Too many users:** The maximum number of concurrent users was reached. Wait a couple of minutes and then try to log in again.
- **The user xxx is already logged in:** The account you want to use is a named account and is already in use. After this message appears, you can use the login button a second time to stop the current session of your account and to re-login again.

Logging out

The logout button



When you are logged in, you can log out at any time by using the "Logout" button at the upper left corner. After logging out, you will be forwarded to the login-dialog again or can use the system as a guest (only if the guest-mode is enabled for your server, see above).

The new web 2.0 version of the instantOLAP web-frontend automatically ends your session if you close the browser or navigate to another site. If you logout from the system you'll need to have to re-login the next time, even if you used the "Remember me" checkbox from the login-dialog.

The portal structure

Portal layout

The portal layout is divided into two main parts: The explorers at the left border and the window manager in the main area.

The **explorers** are used to navigate inside the repository, the models or to use other tools.

The **window manager** is the area in which all content like queries will open. Because it can hold more than one window at a time, all windows are arranged in a tabular order. You can easily change between open windows by clicking onto their tab or by using the window manager menu which is described later.

Navigation and explorers

All reports and other files inside instantOLAP are stored in the repository, a folder based project structure comparable to the file-system of a personal computer. Navigation inside the repository is possible by using the repository explorer.

In addition to the repository explorer there is another tool which can be used for inspecting the repository, the navigator panel. It is a window which is displayed in the main area of the portal and always show a detailed view of the folder currently selected in the explorer.

Beside the repository navigation there are also a number of other explorers which allow to create and manage bookmarks, inspect and query the OLAP models and perform special activities like changing your password etc. Depending on you access rights, you can find one or more of the available explorers here:

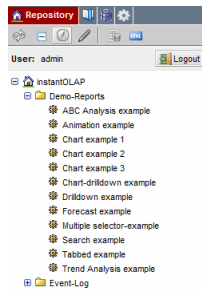
- The repository explorer allows the navigate within the repository and to open and manage reports and other files. This explorer is available for every user.
- The search explorer allows to search for queries inside the repository.
- The bookmark explorer contains all personal bookmarks for a user and allows to edit, arrange or delete them. The bookmark explorer is available for every user unless he is logging in as a anonymous guest.
- The model explorer shows the current OLAP model while editing an adhoc report in the Analyzer. In all other situations this explorer is invisible or empty.
- The tool explorer contains a list of tools which can be used for several purposes, e.g. for changing your password. It is available for every non-anonymous user but contains different tools depending on your account type.

All these tools are placed in the explorer area at the left side of the browser. The are arranged in a tabular view and you can switch between these explorers by clicking onto their tabulators in the very upper left corner of the portal.

The repository explorer

The repository explorer is the most important and first tab of the explorers and shows all folders, queries, snapshots and other files you have access to with your account.

The repository explorer



To view a query, snapshot or pivot-table, simply click onto its row (onto the title or onto the icon left to the title).

To open a folder, click onto the its row or on the minus sign left to it. Because the portal is a web 2.0 application, the folder will open without reloading the web page. If the navigator page is opened (see below), selecting a folder will bring it to the front and it will display the current folder with its files and details.

The toolbar

The repository explorer contains a toolbar with common and some restricted buttons:

Refresh



The refresh button will reload and update the explorer content. Unless you use the refresh button, changes in the repository will not be visible.

Collapse



The collapse button will close all nodes inside the repository tree and bring it back to its originally state.

Open navigator



This button opens or closes the navigator panel, which shows more detailed information about the queries and other files in the currently selected folder (see below).

When the navigator is opened, the button will remain pressed. The browser will remember the state of this button, even if you close it and relogin later.

Edit



For power users and administrators, the edit button allows editing the repository. The behavior of the repository changes by using this button: When single clicking onto a folder or file, you will only select and not open it.

In the edit-mode, all entries inside the repository explorers have context menus, which can be opened by using the right mouse-button or by holding the CTRL key when clicking onto the entry (if using context-menus is disabled in your browser).

Depending on the type of the entry and your access rights, different options will be available which will be described later in this chapter.

Create new pivot table



With this button, power users and administrators can create a new pivot-table inside the browser and store it in the repository later on. The creation and usage of pivot-tables is described in its own chapter.

Create empty HTML page



This button allows to create an empty HTML page. The HTML editor will open in the edit mode and contain the new HTML page.

Context menus

When switching the repository explorer into the edit mode by using the edit button, all items inside the repository tree will have context menus which can be opened by using the right mouse button or by holding the CTRL key when clicking onto it.

Depending on the item type (folder or file) different menus with different items will appear:

Folder menus

Delete folder



This item will delete the selected folder and its complete content recursively. Before the folder is deleted, a confirmation dialog will appear.

Rename folder



Use this menu item to rename a folder. The name of a folder is not the display name you see inside the explorer, this can be changed when editing the folder properties with the folder editor.

Edit



This menu opens the folder editor which allows to change the folder properties and its access rights. The editor is described later in a separate chapter.

Upload



The portal allows you to upload files into folders if you own the necessary access rights. You can upload all types of documents, including HTML and PDF files or queries and pivot-tables. After clicking onto this item, a upload dialog will appear and ask you for the local file to upload.

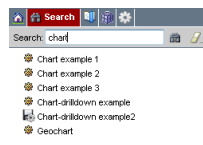
File menus

Files only can be deleted or renamed by using their context menus. The items and their usage are the same as for the folders.

The search explorer

The search explorer is a useful tool which allows to search for queries or other files by their name.

The search explorer



It contains a simple toolbar with a text field and two buttons. Use the field to input your search phrase and press enter or the "Find" button next to the field to start the search.

Now the server starts searching. It will return all files you have at least read access on and which contain the entered phrase in their name. The explorer always searches in the complete repository.

After the search is finished the result will be displayed as a list below the toolbar. Like in the repository explorer, you can open the files with a single click.

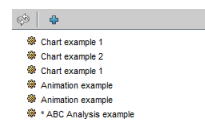
If you want to empty the search result, you can use the last button "Clear" from the toolbar.

The bookmark explorer

instantOLAP contains an integrated bookmark system, which allows to create bookmarks for frequently used reports or documents and to edit their parameters, e.g. the default selection for reports.

Also, you can define bookmarks as auto-start bookmarks which will automatically open when you log into the system.

The bookmark explorer



The bookmark explorer displays all your bookmarks in a list. All auto-start entries are marked with a star before their named, e.g. in the image above the bookmarks "Chart example 1" and "Animation example" are auto-start entries.

You can use a bookmark easily by clicking onto its row (its name or icon). When clicking onto a bookmark, a new window will open in the main area of the portal and show the query, pivot-table or document to which the bookmark points.

The explorer also contains a toolbar and a context menu for each entry:

The toolbar

Refresh



The refresh button updates the list of you bookmarks inside the browser

Add



This button adds bookmark for every window which is currently opened in the main are of the portal. Not every type of window is able to create a bookmark, only queries, pivot tables and documents (HTML, PDF or Excel files) can be stored as bookmark.

The context menu

Each item in the bookmark explorer contains a small context menu with the following items:

Delete



Use the delete item to delete a bookmark from your list of bookmarks. Deleting a bookmark will not delete the original query or file to which the bookmarks points to.

Edit



Use the edit button to change the properties of a bookmark (its name, the autostart-flag and the default parameters for queries). When clicking onto this button, the bookmark editor dialog appears (see below).

Move down and move up



The move up and move down items allow to change the order of your bookmarks by moving single items within the list upwards or downwards.

Toggle auto-load

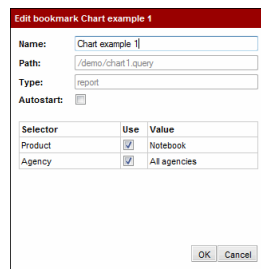


By using the toggle auto-load button, you can change an ordinary bookmark into a auto-start bookmark and vice versa.

The bookmark editor

The bookmark editor allows to change some properties of a bookmark, especially its name, the auto-start flag and the default parameters.

The bookmark editor



The name of a bookmark is its display name which is visible in the list of all bookmarks inside the explorer.

The auto-start flag determines if a bookmark will open automatically when logging into the system. Changing this flag is equal to use the toggle auto-start item from the context menu of an item.

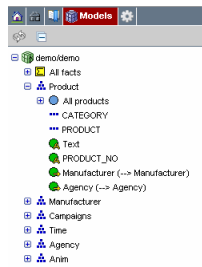
The selector table shows all parameters and their values for queries. You can delete parameters from the list by de-selecting them in the list. If you delete a parameter, the query will use its default value for this parameter when opening it by using this bookmark.

To store the changes simply us the "OK" button of the editor dialog. Using "Cancel" will drop all changes.

The model explorer

The model explorer is only visible for power users and shows the current used model whenever a power users edits a pivot table in the analyzer.

The model explorer



Power users can use this explorer to drag elements (measures, levels, keys etc.) from the model into their pivot tables.

The toolbar

Refresh



This button will refresh the current model.

Collapse

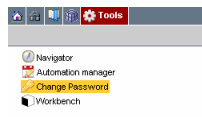


This button collapses the model tree and brings it into its original state.

The tool explorer

The last explorer is the tool explorer. It shows a list of all additional tools available for the current user.

The tool explorer



There is a number of tools which can appear here:

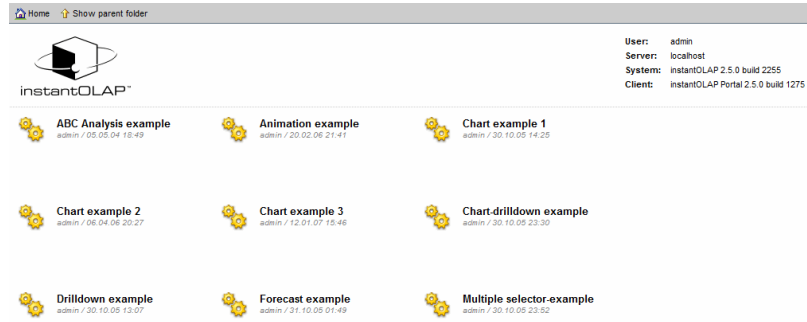
- Navigator: You can also use this button to open or close the navigator panel (this is equal to the navigator button in the repository explorer).
- Automation manager: This tool is available for all non anonymous users and enables them to manage their own background automations, e.g. the subscription of reports by email.
- Change password: Allows the user to change his password. This tool is available for all non anonymous users unless the user management has been disabled or changed on the server side.

You can start a tool by simply clicking onto its row. There is no toolbar or context menu available for this explorer.

The navigator

The navigator is no explorer but it is strongly connected to the repository explorer. The navigator will always display the folder which is currently selected in the repository explorer and provide more detailed information about the queries and files contained in the folder.

The navigator



Beside the folders and files the navigator also show the logo of your server or selected folder and some version-information about your server and frontend.

Opening the navigator

There are two ways to open the navigator: Use the navigator button from the repository explorer toolbar or use the navigator tool inside the tool explorer.

After you opened the navigator, it will always pop up when you select a folder inside the repository explorer.

If you close the browser with opened navigator and re-login later, the navigator will appear automatically (its visibility is stored as a cookie in your browser).

You can close the navigator by simply using the close button of its window or by clicking onto the navigator button in the repository explorer again.

Using the navigator

For each query, snapshot or pivot-table it shows a big icon, the title, its author and creation date (if available) and its description (also only if available). For all other files, only a big icon and the filename is displayed. You can open a file by clicking onto its icon or onto its title.

If you click onto a folder or a folder title, the navigator will open this folder and display its content.

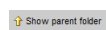
There is also a small toolbar which allows some navigation inside the repository:

Home



The home button moves to the root folder of the repository and displays it inside the navigator.

Show parent folder



The parent button moves up one level in the repository structure and displays the parent folder inside the navigator.

By navigating inside the navigator, the current selection of the repository explorer will no be changed.

The window manager menu

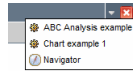
At the upper right corner a small menu appears if any window is opened. It contains a popup menu which allows to switch between the windows and a "close all" button which closes all currently opened windows.

The window manager menu



If you click onto the menu symbol (the little triangle left to the close all button), a popup menu will appear which contains a list of all opened windows with their symbols and titles.

The list of all opened windows



By clicking onto one of these titles, you can display the corresponding window and bring is also to the first position inside the window manager.

Error messages and other notifications

All activities and notifications are displayed as little message boxes at the lower right corner of the display. They inform you about successfully finished operation, errors and the current progress of long running processes.

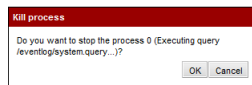
A progress notification



The panels will pile up if more than one message or current progress is displayed. Message panels will disappear automatically after 10 seconds, progress panels disappear when the process is finished or canceled.

By clicking onto a error message panel, you can open a dialog which contains a larger and more detailed message. By clicking onto a process panel, a dialog will open and ask you to cancel this process.

The kill process dialog



If you choose "OK", the process will be killed and the progress panel will disappear.

Using queries

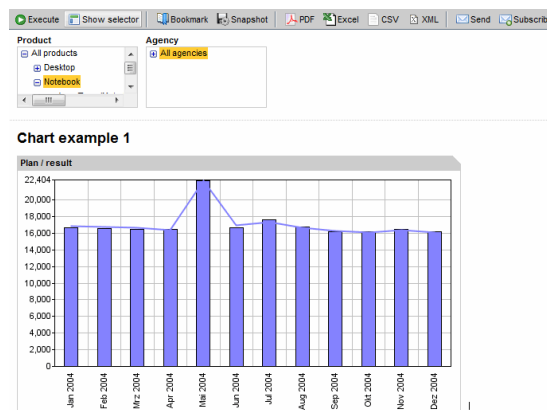
Opening queries

As described in the chapter before, there are different ways to open a query inside the portal: By clicking onto it in the repository explorer, by using the navigator panel, by searching it or by opening a bookmark (which refers to a query).

A query will be opened as a window in the main area of the portal. The name of the window is equal to the name (not to the title) of the query.

The query viewer is divided in different segments: A toolbar at the top of the viewer, the selector bar at the top of the client area and the result which is displayed below the selector bar.

The query viewer window



The toolbar

Most functions of the query viewer are available as buttons in the toolbar. Depending on the query, your user type and your access rights, some of these buttons may not appear.

This is a short overview about the buttons. Most functions will be described more detailed later in this chapter.

Execute



The execute button executes or re-executes the query with all settings which are currently selected in the selector bar. Using this button is not the only way to execute the query, some selectors may trigger also an execution of they have been configured as auto-submit selectors.

Print view



The print view button displays the current query with the current selector settings as PDF file below the selector bar. In this mode, you can immediately print the query by clicking on the print button of the PDF plugin without the need to export the report to a PDF file before.

Note that the selector bar also works in the preview mode and you can still change the report selecting while using the preview.

To stop the preview mode and to return to the normal HTML view simply click onto the preview button again.

Show selector 

Use this button to hide or show the selector bar.

Bookmark 

This button is used to add a bookmark on the current query with its current selection.

Snapshot 

Similar to bookmarks, you can make snapshots from queries and store them in the repository. Snapshots are frozen results which are independent from the OLAP model and can be viewed at any time without the model.

Export to PDF, Excel or CSV 

Use this buttons to export the current result to one of the available export formats (PDF, Excel or CSV).

Send 

You can send the current query result to any email recipient with this button. The result is send as a document, either in the PDF, Excel or CSV format.

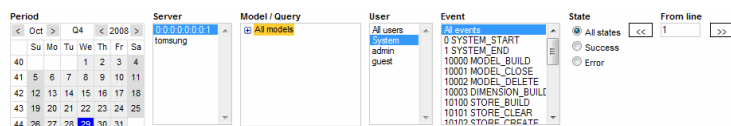
Subscribe 

Instead of sending the query immediately, you can also subscribe it and send it to your email address in a self defined frequency. With this button you will open the automation manager (where you manage all your subscriptions) and a new entry for this query will be added.

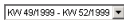

The selector bar





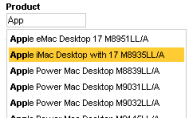
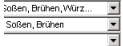
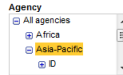
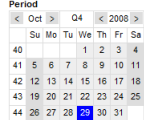
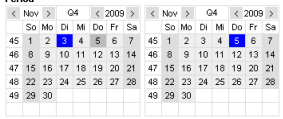
The selector bar allows to influence the current selection of dimensions and variables for the query. With the selection you can determine which data is being displayed in the report (e.g. for which day or product you want to see the data).


A selector bar



The bar varies from report to report and may contain a different number of selectors, each offering different options and with one of this possible selector-types:

<p>Single-Selector</p> 	<p>Allows the selection of a single element from one dimension. The Single-Selector is displayed as a simple combo-box with which the you can select a single element.</p>
<p>Multiple-Selector</p> 	<p>Allows the selection of one, more or no elements from one dimension. It is displayed as a list box where you can select multiple elements from by holding the "Shift" key while selecting the desired elements.</p>

<p>Interval-Selector</p> 	<p>The Interval-Selector lets your select a whole interval of elements by selecting the first (from) and the last (to) element of the desired range. This selector is mainly used for selecting time-intervals.</p>
<p>Radio-Selector</p> 	<p>The Radio-Selector works like the Single-Selector, but the elements are displayed as radio-buttons and are all visible at the same time.</p>
<p>Checkbox-Selector</p> 	<p>The Checkbox-Selector works like a Multiple-Selector, but the elements are displayed as check boxes and are all visible at the same time.</p>
<p>Search-Selector</p> 	<p>With the Search-Selector you can search elements within a dimension by entering a search-pattern. The search-pattern can contain simple characters and digits and wildcards (the possible wildcards are ? and *).</p> <p>The wildcards ? stands for one single unknown character, the * for a number of them. E.g. the pattern "H*e" will search for all items starting with "H" and ending with "e". Note that the search-selector is not case sensitive.</p>
<p>Suggest-Selector</p> 	<p>The Suggest-Selector allows to find a single selection by giving the user suggestions while he inputs text into the text box. The suggestions are all options which contain the current input anywhere in the option text. By using cursor up and down or the mouse, the user can choose from one the offered options.</p>
<p>Hierarchy-Selector</p> 	<p>The Hierarchy-Selector allows the search and selection of a single dimension-element by navigating through the dimension hierarchy.</p>
<p>Hierarchy-Tree-Selector</p> 	<p>The Hierarchy-Tree-Selector works similarly to the Hierarchy-Selector but is displayed as an interactive Tree.</p>
<p>Calendar</p> 	<p>The Calendar-Selector is used to select a day, week, month, quarter or year from the time dimension. The dimension is displayed in form of a classic calendar where you can select an element by clicking on it.</p> <p>Note that you only can select that items which the report is configured for. All other items will be deactivated.</p>
<p>Calendar-Range</p> 	<p>The Calendar-Range-Selector works similar to the normal calendar selector, but it shows two calendars and allows to select a start and end date.</p>

<p>Button-Selector</p> 	<p>The Button-Selector changes the selection to a specific element which is defined by the query-creator (e.g. the previous day or the most selling product).</p>
---	---

Executing the query

To start a query after you set all selectors to their desired values simply click onto the execute button in the toolbar (the first button).

Usually, some or all selectors in the selector are configured to auto-submit their values after you changed the selection. In this case the query will be executed automatically and you don't need to use the execute button.

If you change any selection or re-execute the query while it is currently being executed, the current execution will be stopped and the query starts again with the new parameters. Therefore, even if you change the selectors often and don't wait until the query has finished, you will not run more than one process at a time on the server.

Working with the query

After the query is executed its result is display in the main area under the selector, together with the title and blocks with their tables or charts.

Some of these elements allow further interaction, depending on the query the following operation may work:

Drilldown

Inside tables you can drill down or roll up within the hierarchy of a dimension. Drilldown means to unfold the "children" of an element in order to view the details of aggregated values (e.g. to view the sales of single products belonging to one product group from which you drilled down).

Drilldown must be enabled for a query, otherwise you would be able to use this feature. Drilldown enabled queries contain drilldown and rollup symbols at the left side of the elements, which you can use to unfold or fold the detailed view for one element by clicking onto them.

A report offering drilldown

Quantity / Turnaround by category					
	Quantity	Planned	Amount	Planned	%
▼ Desktop	10,312	10,390	21,062,662	21,594,734	97.54%
Apple eMac Desktop 17 M8951LL/A	2,023	2,016	4,806,000	4,952,755	97.04%
Apple Power Mac Desktop M8639LL/A	2,126	2,164	6,844,365	7,157,713	95.33%
Compaq Presario 6480NX Desktop PC	2,034	2,903	2,030,746	2,143,619	94.73%
HP Pavilion 544n Desktop PC	2,136	2,153	2,408,627	2,459,252	97.94%
Sony VAIO PCV-RS310 Desktop	1,989	1,954	4,972,461	4,881,395	101.67%
▶ Notebook	16,190	16,085	44,128,925	44,477,151	99.22%
▶ Pocket PC	7,897	7,846	4,689,585	4,708,515	99.60%
▶ Tablet PC	6,021	5,969	17,241,914	16,881,080	103.36%
All products	40,420	40,290	87,123,086	87,461,480	99.61%

Depending on the query, the child elements of a header will be displayed under or nested to it.

Links

Table cells (both header and data cells) or charts (e.g. bars) of a query can be linked to other queries in order to allow a different or more detailed view of the data displayed by the regarding header or cell.

When using a link, the target query will be opened with the dimension selection of this cell. E.g. a table cell showing the sales for the product 'A' linked to a trend-analysis will automatically open the trend-analysis for this product 'A'.

Like in HTML, links can be texts or linked icons. In the case of a linked text, you must click onto the cell-text to open the linked query. In the case of an icon, only the icon will be linked and you will have to click onto the icon (displayed at the right border of the cell) to open the other query.

By clicking onto a linked text of a cell or a linked symbol inside a cell you can open the detail-report which is linked to this cell. Depending on the link the new query can appear in the same window (then you can use the back button of your browser to navigate back to the original query) or in a new window.

Depending on the link there may also be a popup-text which appears when moving the mouse over the link. This text can give some additional information about the link target.

Text input

Depending on your access rights and of the current query, some fields of the query may accept data input. If you want to change the cell contents, simply click into the cell and the cursor will become visible inside the cell.

After you change the content of a cell and leave it or press enter, the new value will be send immediately to the server. A little triangle in the upper left corner will signal if the new value was correct and accepted or if an error occurred, e.g. when the format or type of the new value was wrong.

Entering data into
table cells

	Amount	PlannedAmount	Quantity	PlannedQuantity
▶ Desktop	20.709.942	20.998.185	10.352	13.360
▼ Notebook	45.343.858	45.643.653	16.688	17.844
Acer TravelMate 233XV Notebook	5.530.856	5.684.322	2.015	2.997
Acer TravelMate 281XV Notebook				invalid
Acer TravelMate TM233XV Notebook PC				
Acer TravelMate TM234LCI Notebook PC				

A green triangle signals a correct and accepted value. A red triangle means incorrect data. In this case you can correct your input or use the ESC key to drop your input and leave the field.

Analyzing blocks

For power users, blocks contain a context menu which allow to open the current table or chart as a editable pivot-table in a new window.

To open the context menu, locate the mouse pointer above the table or block and use the right mouse button. As an alternative, you can also click onto it while holding the CTRL key (if your browser blocks context menu).

After choosing the menu item, a new window will open which contains only the selected block and all selectors from the original query. You can edit and save this new pivot-table to a new file then.

Exporting queries

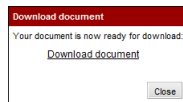
The query viewer offers different options to export queries. You may convert a query result to a document and download it to your desktop or you can send it as Email to any recipient.

Exporting queries as documents

Converting documents and downloading them is a very easy task. Use one of the export buttons from the toolbar to convert the result to either a PDF, Excel or CSV file.

While the conversion is in progress, a notification panel will appear until the server finished the export. Then a dialog will appear offering a download link to the new document.

The download dialog



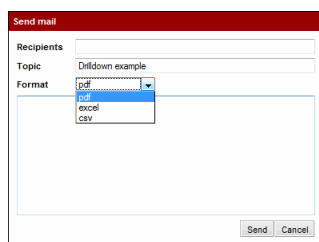
Like in normal HTML pages, you can click onto this link and open the document or you can use the "Save as..." item from the browser context menu and save the document on you computer.

Sending queries

Sending a query is very similar to downloading it, but after the document is generated it will be send as Email. To send a query, use the send button from the toolbar.

Before the sending starts you must provide some information to the server. Therefore, a dialog will open:

The send dialog



You can edit the following properties:

- Recipients: A list of Email addresses to which you want to send the document to. If you want to specify more than one recipient, you must use the comma char as separator.
- Topic: This optional field allows to define the topic of the mail. The title of the query is prefilled as default topic.
- Format: The Email will contain the generated document as attachment. The format of the attachment is defined here, possible options are PDF (default), Excel or CSV.
- Body: The large text field contains the body text of your message. You may leave it empty or add some text here.

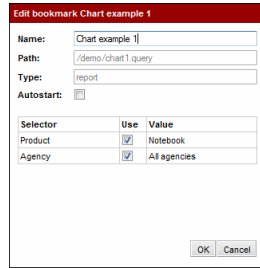
After you completed your input, you can use the "Send" button to generate and send the document. If you use "Cancel", the dialog will be closed and no document is generated.

Creating bookmarks

Instead of using browser bookmarks you can also use the powerful built-in bookmark system of instantOLAP. To add a bookmark for the current query to your list of bookmarks use the bookmark button from the toolbar.

Before the bookmark is added to your list you can change some of its properties and the default selections of the selectors. Therefore a dialog opens after you pressed the bookmark button:

The bookmark editor



You will see the following properties:

- Name: The name of the bookmark as it appears in your list of bookmarks.
- Path: The path to the original query. This property cannot be edited.
- Type: The type of the bookmark, it is set to "report" when bookmarking a query. This property cannot be edited.
- Auto-start: When selecting this checkbox, the bookmark will be opened automatically when you log into the portal the next time.

Below the properties a table with all selectors of the query is displayed. For each selector you can see its name and its currently selected value.

If you don't change any here, the bookmark will open the query with the exact same selection. If you un-check selections within the table, the selectors will open with the default selection defined in the query.

For example, if a query allows to select a product and a date, you may want to remember the selection for the product inside the bookmark but not the date, because the query by default shows the current month. Then the bookmark would open the query with the current month and the product selected by you.

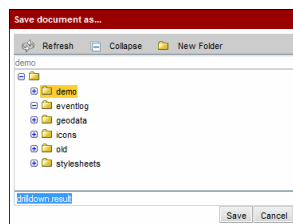
Creating snapshots

Snapshots are "frozen" query result which become independent of the OLAP model in the moment you save them into the repository. Snapshots behave like queries but you cannot change any selection because only the visible data is stored in the snapshot (changing a selection would mean the query viewer had to query the OLAP model again).

You can use snapshots to make copy of the data which will not change in future whenever the original data changes. Also, snapshots can be useful to provide reports to a very large number of users because viewing a snapshot does not trigger any database activity and is very CPU efficient.

To create a snapshot, use the snapshot button from the toolbar. The system will open a file dialog and ask you where to save the file:

The save dialog



The save dialog behaves like a typical file dialog known from desktop computers.

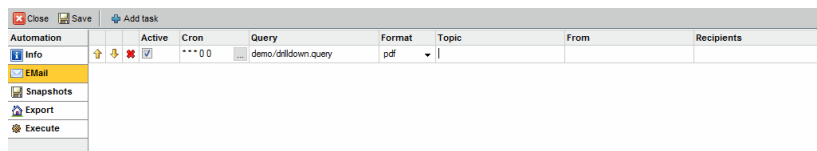
Because the repository has an access control system, you can only place the snapshot into folders you have administration rights to.

Subscribing queries

Instead of sending a query as Email immediately, you can also subscribe a query and let the server send it to your Email account frequently.

Each user owns it list of subscriptions and other automations, which can be edited using the automation manager. When clicking onto the subscribe button in the toolbar, the automation manager will be opened and a new entry for the current query will be added.

The automation editor



You can edit the new entry now, especially the schedule plan and the recipients are necessary entries you should change / add for the subscription. After you finished editing the entry use the save button to activate your new subscription.

The automation manager is description more detailed in the tool chapter of the portal documentation.

Pivot-tables

Pivot-tables behave similar to queries but only contain a single table or chart (in difference to queries, which work more like dashboards and can contain more elements).

Pivot-tables can be edited inside the browser and use a new and flexible viewer, which e.g. allows to scroll inside the table without moving the table headers out of the view or to change the visualization of the table while viewing it.

Because pivot-tables and queries have many similarities, you can easily change between both formats. You can create pivot-tables from a block inside a query or you can publish pivot tables as queries also. The basic difference between both is only the maximum number of elements they may contain.

Opening existing pivot tables

To open an existing pivot table works exactly like opening a query or any other document. Use the repository explorer or your bookmark list and open the table by clicking onto it.

When opening a exiting table it will be displayed in the viewer mode, but you can change every time to the edit mode and change it.

Creating a new pivot table

There are two ways to create a new pivot-table:

- You can create a new table from scratch using the "Create new pivot-table" button in the toolbar of the repository explorer,
- you can convert a block from the query-viewer to a pivot-table by using the context menu of that block.

Remember that you can only create pivot-tables on OLAP models which are started and which allow analyzing them. This has to be configured by your administrator, because not every model is useful for adhoc analytics.

Creating a new table from the repository explorer

The toolbar of the repository explorer contains a button named "Create new pivot-table" which creates an empty table from scratch. Before the system will create a table, it will ask you for the model you want to analyze. A small dialog with a list of all started and available model will appear from which you must select one:

The model selector dialog



After you selected the model use the "OK" button to create the table. A new window will open, containing the empty table.

Creating a new table from the query viewer

Another way to create a pivot table is to use a block from an existing query as a template. Each block in query has its own small context menu with an item "Open in analyzer". When using this item, a new window will open, containing the new pivot table.

This table will contain the single block you used as template and also all selectors from the source query. The selectors are a necessary component of the new table as most blocks depend on the selection.

Note that only queries based on analyzable model offer this context menu in their blocks.

The analyzer

The editor for pivot-tables is called the "analyzer". Its structure is similar to the query viewer of the Workbench.

The analyzer

			Jan 2004	Feb 2004	Mrz 2004	Apr 2004	Mai 2004	Jun 2004	Jul 2004	Aug 2004
Desktop	Apple eMac Desktop 17 M8951LL/A	12289 Obere Str. 57 Berlin	240,193.00	175,015.00	236,572.00	248,642.00	346,409.00	136,391.00	246,228.00	148,...
	68528 Magazinweg 7 Frankfurt a.M.	68528 Magazinweg 7 Frankfurt a.M.	275,196.00	197,940.00	196,741.00	257,091.00	389,917.00	187,095.00	246,842.00	247,...
	88885 Berliner Platz 43 München	88885 Berliner Platz 43 München	206,397.00	176,222.00	172,691.00	216,053.00	382,275.00	213,839.00	216,053.00	264,...
	78563 Adenauerallee 980 Stuttgart	78563 Adenauerallee 980 Stuttgart	111,225.00	288,473.00	333,132.00	305,371.00	540,736.00	317,441.00	276,403.00	235,...
	12289 Obere Str. 57 Berlin	12289 Obere Str. 57 Berlin	276,192.00	261,386.00	317,282.00	328,800.00	549,096.00	233,446.00	282,632.00	337,...
	68528 Magazinweg 7 Frankfurt a.M.	68528 Magazinweg 7 Frankfurt a.M.	243,312.00	323,868.00	397,848.00	374,832.00	688,836.00	266,328.00	220,296.00	237,...
	88885 Berliner Platz 43 München	88885 Berliner Platz 43 München	304,140.00	309,072.00	404,424.00	315,648.00	549,096.00	310,716.00	289,344.00	238,...
	78563 Adenauerallee 980 Stuttgart	78563 Adenauerallee 980 Stuttgart	437,304.00	368,256.00	404,424.00	411,000.00	671,320.00	259,752.00	468,896.00	437,...
	12289 Obere Str. 57 Berlin	12289 Obere Str. 57 Berlin	85,635.00	58,905.00	61,875.00	94,150.00	131,670.00	63,360.00	61,380.00	76,...
	68528 Magazinweg 7 Frankfurt a.M.	68528 Magazinweg 7 Frankfurt a.M.	92,565.00	90,585.00	105,930.00	69,795.00	178,895.00	106,425.00	79,695.00	84,...
	88885 Berliner Platz 43 München	88885 Berliner Platz 43 München	89,595.00	103,950.00	115,930.00	96,010.00	223,740.00	121,770.00	70,260.00	82,...
	78563 Adenauerallee 980 Stuttgart	78563 Adenauerallee 980 Stuttgart	136,105.00	76,725.00	85,635.00	110,880.00	215,325.00	100,980.00	90,080.00	140,...
	12289 Obere Str. 57 Berlin	12289 Obere Str. 57 Berlin	86,729.00	94,346.00	72,864.00	116,814.00	216,820.00	121,302.00	162,908.00	69,...
	68528 Magazinweg 7 Frankfurt a.M.	68528 Magazinweg 7 Frankfurt a.M.	85,556.00	133,608.00	82,040.00	138,882.00	260,770.00	101,864.00	181,880.00	133,...
	88885 Berliner Platz 43 München	88885 Berliner Platz 43 München	107,238.00	125,404.00	113,684.00	92,002.00	250,222.00	86,728.00	182,246.00	92,...
	78563 Adenauerallee 980 Stuttgart	78563 Adenauerallee 980 Stuttgart	99,620.00	121,302.00	117,786.00	92,002.00	257,254.00	101,378.00	192,794.00	123,...
	12289 Obere Str. 57 Berlin	12289 Obere Str. 57 Berlin	274,365.00	171,990.00	196,560.00	252,525.00	439,530.00	218,400.00	219,765.00	158,...
	68528 Magazinweg 7 Frankfurt a.M.	68528 Magazinweg 7 Frankfurt a.M.	296,205.00	326,235.00	244,335.00	192,465.00	522,795.00	331,695.00	305,760.00	226,...
	88885 Berliner Platz 43 München	88885 Berliner Platz 43 München	350,805.00	212,940.00	245,700.00	252,525.00	554,190.00	309,855.00	274,365.00	343,...
	78563 Adenauerallee 980 Stuttgart	78563 Adenauerallee 980 Stuttgart	334,425.00	311,220.00	279,825.00	348,075.00	683,865.00	251,180.00	318,045.00	215,...
	12289 Obere Str. 57 Berlin	12289 Obere Str. 57 Berlin	210,309.00	190,414.00	269,884.00	204,624.00	497,350.00	166,257.00	271,411.00	285,...
	68528 Magazinweg 7 Frankfurt a.M.	68528 Magazinweg 7 Frankfurt a.M.								

- At the top of the editor the toolbar offers operations for manipulating and managing your pivot table.
- Below the toolbar the selector bar is visible. Depending on the edit-mode, it shows the selectors itself and/or their definitions (the grey boxes with the drag-symbol at the left). The definitions can be copied, arranged or deleted using the mouse and drag&drop.
- In the edit mode, a schematic view of the table is displayed below the selector bar. Like the selectors, the table-definition can be arranged using the mouse.
- In the main area the table itself is displayed, either as scrollable table or as chart. This area will refresh every time you change the selector or table definition.

The toolbar

The analyzer contains a toolbar with several options which can be used to view or to edit the pivot-table:

Execute



Use the execute button to re-execute the query and to rebuild the table at any time. Using this button is not the only way to execute a pivot-table: Like in queries, the table may also execute automatically when changing a selection. Also, it will always execute automatically whenever anything is changed and the data-view of the table is switched on (see below).

Save



The save button stores the pivot table to its original position in the repository and replaces the old version. If the pivot table is new and has never been saved before, this button behaves like the "Save as..." button and opens a file-dialog (see below) before storing the table.

Save as...



Use this button to save the table with a new filename or location. A file dialog will appear and ask you for the new filename and for the folder the table will be stored in. After saving with a new filename, every future save will store the pivot table at its new position.

Publish



You can publish pivot tables as normal queries to make them available for non power users or to edit them with the normal query-editor of the Workbench.

Like when using "Save as...", a file-dialog will open and ask your for the new location of the query.

Bookmark



Bookmarks can also be created for pivot-tables (if they have been saved before). Adding a bookmark for pivot tables works exactly the same ways as for queries.

Show selector



If this button is activated (its initial state), the selector bar is visible. By using this button you can make the bar invisible or visible again.

Display as table /
display as chart

Pivot-tables can be visualized either as table or chart. With this two buttons you can change between both formats. When changing the format, the toolbar also changes because different buttons and tools are available for each format.

Export to PDF, Excel
or CSV

Like queries you can also export pivot-tables as PDF-, Excel- or CSV-documents. The exported document uses the same visualization format you currently see (table or chart).

Send



Pivot tables can be exported and sent as Email directly out of the viewer. When using this button, an additional dialog will open and ask for all necessary parameters to send the document.

Rotate



This functions rotates the pivot-table by swapping its X- and Y-axis. It will also swap the borders and padding of all headers and the "Suppress Rows" and "Suppress headers" settings of the table.

Suppress empty rows
/ columns

Both of these button are stateful and can be activated or deactivated to suppress empty rows or columns from the table body. Suppressing empty rows or columns means that all rows (or columns) with only empty cells in the data-area will be removed from the result, including the headers related to this rows (or columns).

Add sum row / column



This button adds a simple summary row or column to the pivot-table which shows the total of all columns or columns in a additional row or column. The generated elements are independent of their position, e.g. after rotating a table a summary row will turn into a valid summary column and vice versa.

You can combine multiple totals, because a summary element does not include the values of other summaries in its calculation. Therefore you could e.g. create a subtotal using a nested summary row and a total with a final row.

Edit



Enables or disables editing the selector bar and the pivot-table itself. In the edit mode, all definitions for the selectors or the table will be displayed as draggable grey boxes above the selectors and in the schematic view of the table.

Show / hide values



This button activates or deactivates the display of the values and selectors. You can use this button to edit pivot tables without displaying the data after each change.

Clear



This button clears the complete pivot table and its selectors. The table will not be deleted unless you confirm the confirmation-dialog which appears after using the button.

Trashcan



The trashcan is rather a drag&drop target than a button. You can drag any item of the table into in the trashcan in order to delete it from the table. Note that there is no confirmation request when deleting elements this way.

Additional toolbar entries for charts

The following buttons are only available in the chart format:

Display as barchart /
column-chart /
line-chart / spline /
pie-chart



This buttons determine the chart format and will change it immediately after using one of them. When changing the display format it might be necessary to rotate the table using the "Rotate" button to get useful output.

Toggle 3D Mode



Switches the 3D visualization of the charts on or off. Every chart type can be either display as 2D or 3D chart.

Working with pivot tables

Like queries, pivot tables allow a number of different actions which can be performed on them, including:

- Drilldown
- Data input
- Export to PDF, Excel or CSV
- Creating bookmarks

- Creating snapshots
- Sending pivot tables via EMail

All this function have the same behavior for pivot tables as for queries. A detailed description of these action is available in the previous chapter "Working with query" of this manual.

Context menu for pivot table headers

In difference to normal queries, Pivot tables offer a context menu for their headers. This menu allows to sort tables and to change the number of their fixed rows or columns.

Context menu for pivot headers

	Amn
09 Obere Str. 57 Berlin	240,1
28 Maj	75,1
05 Ber	35,1
03 Adl	11,1
09 Obd	75,1
28 Maj	43,1
05 Ber	34,1
03 Adl	17,1
09 Obere Str. 57 Berlin	85,6
28 Mainzwe 7 Frankfurt a.M.	92,6

Sorting

To sort a table, open the context menu of the header you want to sort the table by. The menu offers three different sort items: You can sort the table ascending by the row (or column) content, you can sort it descending or you can remove all sort settings from the table.

Fixed rows and columns

When scrolling a pivot table, all header from the x and y axis are fixed and do not scroll together with table body. The number of fixed rows and columns is automatically determined by the depth of the x or y axis.

You can change the number of fixed rows and columns by using the context menu of a header. Use the "Fix this row" or "Fix this column" item, to fix all rows from the corner to the selected row (or column). To reset the fix rows or columns to the default use the "Remove fixed row" or "Remove fixed column" item.

Editing pivot tables

Query concepts

Pivot-tables are based on the same concepts as queries, but are a simpler version of them. Before you create pivot tables, you should become familiar with the basic concepts of models and queries. Therefore it is recommended to read the model- and query-description from the "concepts" chapter of this manual before creating complex tables.

The edit mode

Before you can edit a pivot-table, you must turn the analyzer into the edit mode by using the "Edit" button. If you create a new pivot-table, the analyzer will automatically start in the edit mode.

The edit button



While the analyzer is in the edit mode, you can see all selector- and header- definitions above the selectors and a schematic table view below the selector. The definitions and headers are displayed as grey boxes with a small drag-symbol at their left border.

Changing axis definitions

Each pivot-tables own two axes, the X- and the Y-axis. The contents of these axes defines, which dimensions or data will be displayed in the table - they can contain a simple level from a dimension, a fact or nested headers with complex formulas.

Every change you make on a axis definition will immediately become visible in the result area. If you want to switch of the permanent actualization of the result you can use the "Display body" button from the toolbar to turn this off.

Adding headers to an axis by drag and drop

The common way to add a new header to an axis is to drag an element out of the model explorer left to the analyzer. You can drag dimension-levels, single keys, attributes or other expression from the model into the axis. You can also drag existing elements from your pivot table like headers or selectors to other locations.

When dragging an existing element, this will **move** it unless you hold the CTRL key pressed while dragging. When dragging elements from the model explorer, this key has no effect.

When dragging an element into an axis, the exact position where you drop it has an important effect on the table:

- If the axis is empty, it will only show an empty field with the text "Drop elements here" - just drop the element anywhere inside the field and it will become the first header in the axis.
- If the axis already contains one or more headers, you must choose where to place the new element. You can place it on any border of a existing axis-element. The position will determine the position of the new header: It will appear before or behind the other header if you drop it at the top or bottom border in the Y-axis (or at the left or right border in the X-Axis). Or it will be nested under (or become the container of the existing header) if you drop it on the right or left border in the Y-axis (or at the bottom or top border in the X-Axis).

If you drag a new element "behind" or "before" an existing header, this simply determines the position of the new element. E.g. if you drag a fact at the right border of the last element in the X-axis, this fact will become the new last column of your table.

If you "nest" headers, this can have two different effects:

- If the nested header is based on the dimension as its owner, the owner will influence the content of the new header. E.g. if you nest a level "PRODUCT" from a product-dimension under an existing header showing the "CATEGORY" level of the same dimension, the nested header will only show the products of the category which is display in the owner.
- Otherwise, the nested will repeat its whole content for each generated header of the owner. E.g. when nesting a "MONTH" level under the "CATEGORY" level, each month will be displayed for each category as there is no dependency between them.

Especially when nesting non-related elements, it might be useful to use the "Suppress rows" or "Suppress columns" button to eliminate the combination of dimension keys without results.

Adding headers to an axis by using the corner context menu

The corner of the table definition offers a context menu which allows to add an empty header to the x or a axis of the table.

The corner context menu



Use this items if you want to create an empty header without dragging an element from the model into the table.

Adding summary rows and columns

Another way to create new headers is to use the "Create summary-row" or "-column" button from the toolbar. This will create a new header with a predefined formula and add it to the end of the Y- or X-axis. After creating the new header you can also move it to a different position or axis.

It is also possible to create more than on summary header in an axis and to combine them, e.g. when you want to create a subtotal and a total in the same axis. You should always create new summary headers with the toolbar instead of copying existing ones, because each generated headers creates a unique summary variable and copied headers can interfere if the use the same variable

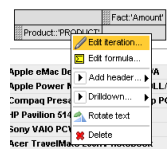
Rearranging headers

You can always rearrange headers and change their position with drag&drop. You can also drag headers into the selector bar and create a new selector with the header-options as iteration.

Context menus for headers

Each header owns a context menu which offers some important operations. To open the context menu, use the right mouse button or hold the CTRL key while pressing the left mouse button on the header (some browsers do not allow the usage of context menus for security reasons).

The header context menu



The following items are available for headers:

Edit iteration: This item will open the expression editor and lets you editing the "iteration" expression of the header. The iteration defines which particular rows or columns are generated by this header, e.g. on row per product or one column per product.

By dragging an element from the model explorer, you create a new header which own the element you dragged from the model as its iteration expression. Changing the iteration with the expression editor can become very complex and is recommended for experienced users only.

Edit formula: Like the iteration of a header its "formula" can also be edited with this item. In difference to the iteration, the formula does not define the number of rows or columns a headers generates but the values which are displayed in the body below or right to the headers.

Formulas are an advanced feature of pivot tables and are only necessary if the calculation you need is not covered by any fact, e.g. if you want to display the difference of two facts. Usually, the formula of headers is empty and the displayed value is only determined by the facts you dragged into your table.

Disable/enable drilldown: This item enabled drilldown for the selected header. A header with drilldown allows to "unfold" its generated header in order to see the children of the unfold key. If the displayed key has no children, there will be no visible drilldown button in the result header.

Add header: Use this item and its sub items to add new and empty headers before, behind or nested under the header. Like the context menu of the corner, this will create empty headers without iterations.

Rotate text: To save space you can rotate the text of headers by 90 degrees. By using this item again you can also display already rotated texts horizontally again.

Delete header: Use this item to delete a header from an axis. This will only delete the selected header, all other elements in the axis remain.

Editing the selector bar

Editing the selector bar works similarly as editing the pivot table: Selectors are also displayed with a definition header at the top and a "body" (the selector itself) at the bottom.

Like headers, selectors can be created, rearranged or deleted using drag&drop. You only can drag its definition - if you use the mouse on the selector-body you will change the current selection of the report.

The header is also displayed only if the analyzer is in the edit mode. An the selector itself will only be display if the "Display body" button in the toolbar is activated.

Creating new selectors

Like creating new headers, the simplest way to create new selectors is to drag an element out of the model explorer left to the analyzer.

If no selector exists, the only place you can drop an element is the "Drop elements here" field of the selector. Dropping an element here will create a new selector at the end of the selector bar. This field is also visible, if there already exist other selectors.

If other selectors exist, you can drop elements at their left or right border and create new selectors between them.

The type of the new selector depends on the element you drop:

- If you drop a dimension-node (the topmost elements in the model-explorer), this will create a new hierarchy-selector which allows to select a single key of the dimension in a tree-like widget.
- If you drop a dimension-level, this will create a single-selector (a simple combo box) which allows to select a single key of that level.
- If you drop a single key in the selector bar, this will create a constant selector which always selects this keys.

After you created a new selector, you may change its type and options using the context menu of the selector.

Rearranging selectors

You can always rearrange selectors and change their position with drag&drop. You can also drag selectors into a table-axis and create a new header with the options of the selector as new iteration.

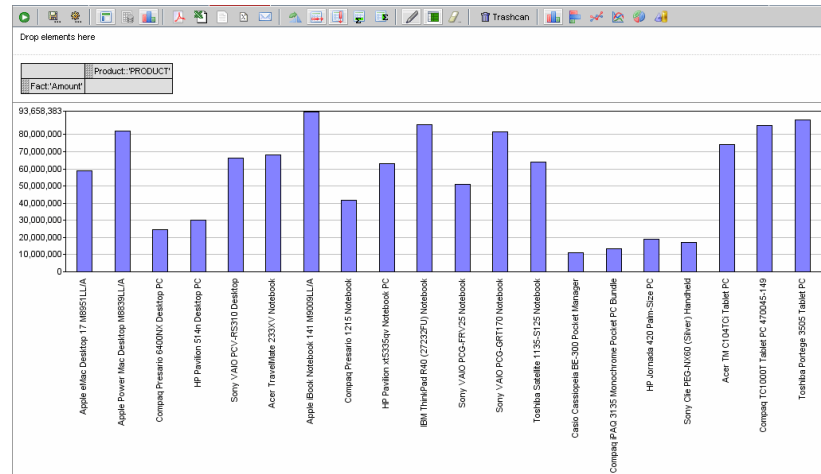
Charts

Pivot tables can also be displayed as business charts by changing the display format of the analyzer.

Changing the display type

If you want to change from the table view to the chart view use the "Display as chart" button from the toolbar. After changing it, the pivot-table will instantly be visualized as chart and a couple of new buttons appear in the toolbar.

The chart view



The chart is always displayed with the maximum possible size which fits into the analyzer window. If you want to enlarge the image, you'll have to resize the browser window or to hide the selector bar.

Chart types

There are five available basic-types for displaying a pivot-table as a business charts:

- **Bar charts:** Each column of the pivot-table becomes a single bar in the bar-chart format. If there is more than one row, each row will be converted into a series of bars which are displayed next to each other (or in 3D charts behind each other).
- **Column charts:** A column chart is just a rotated bar chart and the bar orientation is horizontally. Even if the column chart is rotated, the input data must have the same format as for normal bar chart (each column becomes a bar, each row becomes a series).
- **Line charts:** Line charts display each row from your pivot-table as a line. Each column of the table becomes on point in the lines.
- **Spline chart:** Spline charts work equal to line charts, but the lines are visualized as interpolated splines.
- **Pie charts:** This format displays a row of your pivot-table as a pie chart where each column becomes a segment of the pie and the segment size is proportional to the cell value. If your table has more than one row, the chart will display multiple pies.

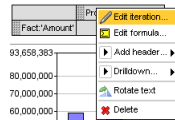
You can change between these types by using the buttons "Display as bar chart", "...as column chart", "...as line chart", "...as spline chart" or "...as pie chart". It is always possible to change the chart type without losing any setting etc.

You can also turn all chart types into 3D charts by using the "Toggle 3D Mode" button from the toolbar. By using this button again, 3D charts are converted into normal 2D charts again.

The expression editor

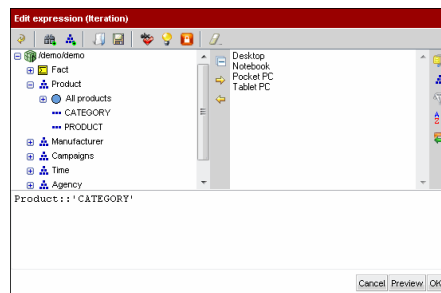
Whenever you want to edit complex expressions like the iteration or formula of a header or the options of a selector, you must use the expression editor dialog. You can open the editor for all editable elements by opening their context menu and using its edit items. In the current version 2.6 header and their iterations and formulas are editable and selectors and their options and default values are editable.

Opening the expression editor for a header iteration



After you opened the editor for a expression, a new and modal dialog will appear. The dialog shows the current expression (e.g. when you edit an existing iteration of a header, the editor will show the current iteration of the header) and a number of elements and buttons.

The expression editor



The editor is divided into 4 main parts: The toolbar at its top, the model tree at the left side, the preview at the right side and the text field which contains the current expression at the bottom.

Adding or removing elements to / from the expression

There are three basic ways to add elements to your expression: You can use the model tree at the left side of the dialog for a fast access on the model items or you can use the dialogs "Find dimension elements" and "Select elements from model".

Adding elements from the model tree

The tree in the editor dialog is similar to the model tree visible in the model explorer when using the analyzer. It shows all dimensions, their levels and their keys.

By double clicking a level or key you can add it to the current expression. As an alternative you can also select an element and use the "Add" button from the toolbar between the tree and the preview list.

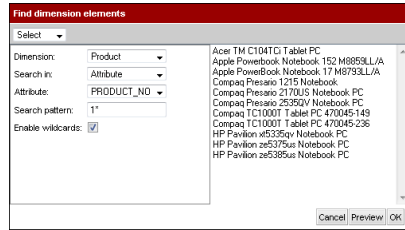
Searching keys

Find dimension elements



Use the "Find dimension elements" dialog to search keys by their ID, display text or attribute values.

The find dialog



The first option box defines in which dimension you want to search for keys. Note the dialog will always search in the entire dimension, not in a specific level only.

The second box lets you choose whether you want to find keys by their IDs, display text or a text attribute. If you choose "Attribute" here, a third option box appears and offers all text attributes of the dimension. Other attributes, like links to other dimensions, will not be visible.

The following input box holds the search pattern. Depending on the check box "Enable wildcards", you can use the wildcards * (any characters) and ? (and single character) here.

You can use the "Preview" button to create a preview of the result at any time. Note that the preview also depends on the select mode in the toolbar.

The select type



Like a couple of other dialogs in the expression editor, this dialog contains a select mode option in its toolbar. It lets you choose to replace your current expression by this, new expression or to modify your current expression with the new one:

- **Select:** This option replaces the current expression by the new expression defined by this dialog.
- **Add:** The Add option adds the new expression to the current expression using the JOIN operator
- **Keep:** Use this option to intersect your current expression with the result of this dialog.
- **Remove:** This option removes the result of this dialog from your current expression.

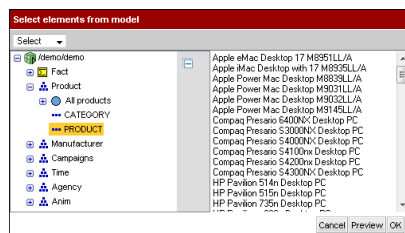
Adding elements from the model

Select model elements



The model dialog is similar to the initial panel of the expression editor. It shows the model tree again and allows to add a single level or key from a dimension to you expression. Simply select an element in the tree and the dialog will create a preview for the expression instantly.

The model dialog



In difference to the basic panel and like the find dialog, this dialog also offers a "Select" / "Add" / "Keep" / "Remove" option and therefore also allows to e.g. remove model elements from your current expression.

Loading and saving expressions

Users can store expressions on the server and reuse them at any time. If you need similar expression often, you can use this feature to save time by only defining the same expressions once.

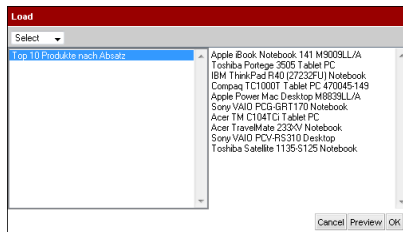
Loading expressions

Load expression



Use the "Load expression" button from the editor toolbar to show the list of the expression you saved in the past.

The load dialog



The list at the left side of the dialog contains the names of all stored expression. You can load the expression by selecting its name and pressing the "OK" button.

You can also create a preview for the expression before loading it. To execute the expression, select its name in the expression list and use the "Preview" button.

Like the dialogs "Find dimension elements" or "Select model elements", the load dialog also to specify the usage of the expression with the drop down list in its toolbar.

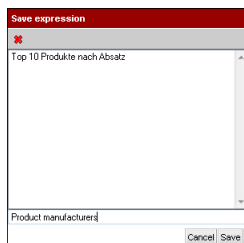
Saving expressions

Save expression



To save your current expression use the "Save" button from the editor toolbar. This will open the save dialog.

The save dialog



The body of the dialog shows the list of all expressions you saved in the past. Below the list, there is a text box where you can input the name of your new expression.

To save the expression, enter the new name in the text box and use the "Save" button from the dialog.

You can also use this dialog to remove existing expression. Simply select the expression and use the "Delete" button from the toolbar. If you want to replace an existing expression, you must select it first, then delete it and finally save your new expression.

Expression tools

Checking the expression

Check expression



Use the "Check expression" button from the editor toolbar to check your expression for its correct syntax and return type. If any error is found in the expression, the dialog will show an error after using this button.

The return type of the expression is often limited by the usage of the expression. E.g. a header iteration can only contain expression which return dimension keys.

Optimizing the expression

Optimize expression



instantOLAP contains an internal expression optimizer which is used every time an expression is executed. You can also execute the optimizer manually here with the "Optimize" button.

The optimized expression is often easier to read than the original expression because the optimizer eliminates unnecessary code.

Freezing the expression

Freeze result



When build an expression with the expression editor, the result is usually a dynamic expression which can return different results depending on its point of usage and on the database content. E.g. if you create a top 10 expression, which returns the best performing products, the result will change whenever the turnaround of the product changes.

If you want the expression to always return the same result in the future, you can freeze the result with the "Freeze result" button from the toolbar. This button executes the expression and converts into a list of constant values, independent on any database content.

Clearing the expression

Clear expression



Use the "Clear" button to clear the current expression. If you cleared the expression by accident you can recover it using the "Undo" button from the toolbar.

Modifying the result

The toolbar right to the preview list contains buttons for a number of modifying tools. The tools allow to modify the current result by e.g. limiting or filtering it.

Limiting the result

Limit



The limit dialog allows to limit the result of the current expression to the first or last X elements.

The limit dialog



The option box allows to select whether you want to limit the result to the first ("Select first X elements") or last ("Select last X elements") elements.

In both cases, the following number field allows you to define the number of element you want the result limit to. If the result contains less elements than you specify here, it will remain unchanged.

Use the "Preview" button to perform a preview on the result and the "OK" button to confirm your changes and to manipulate your expression.

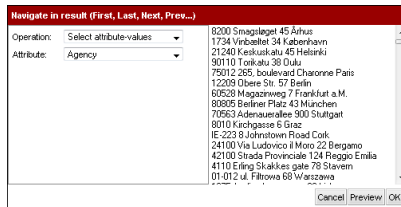
Navigating within the result

Navigate in result



The navigate dialog allows to navigate within the dimension hierarchy of the result keys. Here you can e.g. display the children or parents of your current result or use attributes.

The navigate dialog



The first option box allows to choose the direction of your navigation. There are several options:

- **Select attribute values:** Keys can own attributes which are simple values (like texts or numbers) or relations to other keys. Use this option to show the key attributes.
- **Select children of elements:** Show the children of all your current result keys. Each key may have no, one or more than one child.
- **Select family of elements:** The family of a key is the key itself, its children, the children of its children and so on.
- **Select leafs of elements:** The leafs of a key are only its family members without children.
- **Select parents of elements:** This operation returns the parent of the result keys. Except the root key of a dimension, every key has exactly one parent.
- **Select ancestors of elements:** The ancestors of a key are its parent, the parent of the parent and so one.
- **Select following elements:** Selects the following elements of the result keys.
- **Select previous elements:** Selects the previous elements of the result keys.

Only if you select the "Attribute values" option, a second option box will appear and let you choose the attribute.

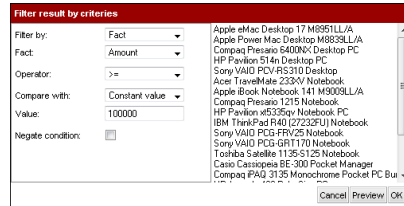
Filtering the result

Filter



Filtering the result means to limit it to list number of keys which match the criteria you define in this dialog.

The match dialog



At first you have to decide, which elements of the keys you want to match. There are four different elements you can use for your filter:

- **ID:** Allows to filter keys by their IDs.
- **Display text:** Allows to filter keys by their display text. The display text is the text which is shown whenever a key is displayed in a table and can be different to the key ID.
- **Key attribute:** Keys can own attributes which are simple values (like texts or numbers) or relations to other keys. If you want to filter keys by one of their attributes, use this option.
- **Fact:** Allows to filter keys by the value a fact returns in the context with the keys.

Depending on the selection you make in this option box, a second option box may appear which allows to select the attribute or fact you want to filter the keys with.

Now you have to select the operator you want to use in your filter criteria. There is list of possible operations:

- **EXISTS:** This operator only returns keys which contain the selected element (ID, text, attribute or fact value). The element value must exist and must not be NULL.
- **IS NULL:** This operator tests the element value and only matches for keys where the element value is NULL. This is the opposite operator EXISTS.
- **IS EMPTY:** This operator only returns keys which contain **no** value for the selected element, even not NULL.
- **=, <, <=, >, >=, <>, LIKE:** These operators allow to compare the element value with a second value. The LIKE operator is used for text comparison and allows to use wildcards (* and ?).

If you have selected one of the operators =, <, <=, >, >=, <> or LIKE, a new option box will appear and let you choose the second value for the comparison. Like the first option box, you can choose between ID, display text, key attribute and fact here. Additionally, the box contains the option "constant value".

If you choose the "Constant value" option, a text box will appear and you can input the value here. If you choose "key attribute" or "fact", a option box will appear and let you choose the attribute or fact.

The last option you have for your match is to negate the result. If you check the checkbox "Negate result", the dialog will only return the keys which **not** match your defined criteria.

You can use the "Preview" button at the bottom to preview your expression and its result at any time.

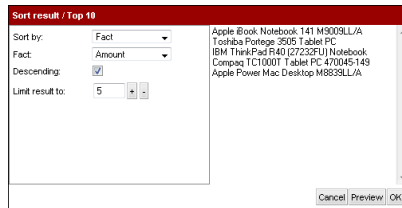
Sorting the result

Sort



Use the "Sort" button from the preview toolbar to open the sort dialog. This dialog allows to sort the result by a couple of different attributes, e.g. by fact values, key attributes or texts.

The sort dialog



The first option box allows to select the element you want to sort the keys by. There are four possible sort elements:

- **ID:** Sorts the keys by their ID.
- **Display text:** Sorts the keys by their display text. The display text is the text which is shown whenever a key is displayed in a table and can be different to the key ID.
- **Key attribute:** Keys can own attributes which are simple values (like texts or numbers) or relations to other keys. If you sort by an attribute which is an relation, the keys are sorted by the display text of the referenced key.
- **Fact:** Sorts the keys by the result value of a fact.

Depending on the selection you make in this option box, a second option box may appear which allows to select the attribute or fact you want to sort the keys by.

By default, the keys are sorted ascending. Use the checkbox "Descending" to change the sort order from ascending to descending.

The last input field allows to limit the number of result elements when sorting keys. By default, this field contains "0" which means there is no limitation for the result size. If you set this field to a value greater than zero, you limit the result to the first n elements.

The combination of the descending checkbox and the limit field allows to generate top 10 queries. E.g. if you sort a list of keys by a fact, set the sort order to "descending" and limit the number of result keys to 10, you will get a top 10 list of the keys for the selected fact.

Like all sub dialogs, the sort dialog allows to preview your expression with the "Preview" button.

Reversing the result

Reverse



This is the simplest modification on the result: By using the "Reverse result" button of the preview toolbar, you can reverse the result of the expression.

Working with documents

instantOLAP is able to display other documents than queries or pivot-tables in the browser. You can also open and view HTML files or PDF documents in the portal. In the case of HTML files you're also able to edit these files if you have the necessary access rights.

Opening documents

Opening a document work same as opening queries or pivot-tables. Just locate the document in the repository browser and use a single mouse click to open a new window containing the document.

Creating new HTML pages

Empty HTML pages can also be created within the repository browser. Use the button "Create empty HTML page" to open a new window with an empty document.

Unless you save the document, it has no filename or location in the repository. So after the first creating of a HTML document you always have to use the "Save as..." button of the toolbar to save it.

Editing HTML pages

If the currently viewed document is a HTML page and if you own the necessary access rights to edit a document (you must be a administrator of a folder to store HTML files or to edit documents contained in that folder), you can turn the viewer into the edit mode.

The edit button



To edit a document use the "Edit" button from the toolbar. In the edit mode, the toolbar contains a number of new buttons, which mainly are used for styling the document:

Stop editing



When the document viewer is currently in the edit mode, you can use the "stop" button to stop editing and to return in normal viewer mode.

Save



After you changed an existing HTML document you can save it to its current location by using this button. The previous version of the document gets lost when storing it.

Save as



Use the "Save as" button to save a new HTML document or to save an existing document under a different name or in a different folder. When using the button, a file dialog will open and ask you for the new name and location.

The alignment buttons



Use this button to align the current paragraph or selection of your text. Possible alignments are left, centered, right and justified.

The font decoration buttons

Use the font decoration buttons to change the text of the current paragraph or selection into bold, italic or striked.

The paragraph styles

With this buttons you can convert the current paragraph into a ordered or unordered list.

The indent buttons

Use this buttons to indent or un-indent the current paragraph.

The horizontal bar

This button will insert a horizontal ruler at the current position of the cursor.

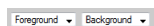
Adding and removing hyperlinks

If you want to link the currently selected text with a URL, you must use the "add link" button. After using the button, a dialog will open and ask you for the hyperlink URL.

If you want to remove a hyperlink from your text, you must select the text and use the "remove link" button. All links from the selection will be remove then.

The font selectors

Use this selectors to choose the font-family and font-size of the current paragraph or selection.

The color selectors

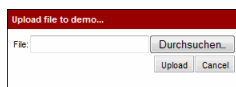
This selectors can be used to choose the foreground or background color of the current paragraph or text selection.

Uploading documents

HTML files and PDF documents can also be uploaded from you local computer into a folder if you own the necessary access rights for the folder.

To upload a document, turn the repository explorer into its edit mode by using the "Edit" button of the repository toolbar. Then you can open the context menu of the folder and use the "Upload..." item to initialize the upload.

A new dialog will appear and enable you to choose the local file you want to send to the server.

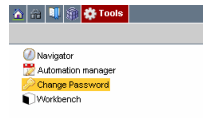
The upload dialog

Use the "Choose file..." button and select, using the following file dialog, the local file. After you successfully selected it, you can use the "Send" button of the upload dialog.

Tools

The tool explorer (the last tab in the list of explorer on the left hand side of the portal) shows a list of tools for you usage:

The tool explorer



The Automation manager

The automation manager is a tool which enables you to configure your personal automation tasks. To use "automation" in instantOLAP means to configure task on the server which are executed on the sever, automatically n the background and with a configurable time schedule.

There are a number of different tasks you can schedule. Depending on your access rights and owned user roles, you will be enable to create a different number of these tasks:

- The "EMail" task automatically exports queries or pivot-tables as PDF, Excel or CSV documents and sends them via EMail to freely definable recipients.
- The "Snapshot" task executes a query and saves the result as snapshot in the same folder the query is located in. Viewing snapshots is much faster than executing a report. This task is only available for administrators.
- A "Export" task exports a query or pivot-table as a PDF, Excel or CSV document onto the file system of the server. This task is only available for administrators.
- The "Execute" task just executes a query. This task is useful when a cache is used in the model and executing the query fills the cache. This task is only available for administrators.

The toolbar

The automation manager contains a small toolbar with the following buttons:

Save



Use this button to save the last changes to your automation table. After saving, all active automation will be executed using their time pattern.

Add task



This button creates a new automation task. The type of the new task depends on the currently opened tab.

Basic settings for the automation manager

The first tabulator, the info tabulator, of the automation manager contains the basic settings for your automation table.

The info tabulator

The both options you can change here are the active flag and the password, which is used for the automation.

If the active flag is deactivated, the complete automation for your user account will be deactivated. This flag is initially activated.

The password field contains the password used for your automation. Because the automation start sessions on the backend server, you must provide your password for an automation. Whenever you create a first automation, this field will automatically be filled with your current password.

Editing EMail tasks

Click on the "EMail" tabulator in the automation manager in order to edit your EMail automation or to create a new task.

The EMail tabulator

Automation	Active	Cron	Query	Format	Topic	From	Recipients
	<input checked="" type="checkbox"/>	****0	demo/chart2link	excel	EMail Test	info@instantolap.com	info@instantolap.com

The table shows all EMail tasks which are stored in your automation table. For each task you can edit the following options:

- **Active:** If this checkbox is selected, the EMail task is activated and will created EMail with the time pattern defined in the Cron column.
- **Cron:** The Cron column controls, when and how often the EMail will be send. To edit the cron pattern, you can use the edit button behind the pattern which will open the cron editor.
- **Query:** This mandatory field contains the path of the query, pivot-table or query-link you want to export and send. The path must always be absolute to the root folder of the repository and must also contain the extension of the file (".query", ".pivot" or ".link").
- **Format:** This field defines the format of the exported file, which will be attached to the EMail. You can send the query as PDF, Excel or CSV file.
- **Topic:** This optionally column contains the topic of the generated EMail. If no topic is defined, the server will use the query name as topic.
- **From:** The also optionally column defines the sender address for the EMail. If no sender is defined, the default sender of the EMail server is used.
- **Recipients:** Use this field, to define or or more recipients for the EMail task. If you want to send the mail to more than one recipient, you must concatenate their EMail addresses using comma.

If want to change the content of the columns, just click into the fields and change their content. There are also three small buttons at the beginning of each row, they can be used to arrange or delete single tasks.

If you want to create a new EMail task, use the "Create task" button from the toolbar. This will add a new and empty task at the end of the task table.

Editing Snapshot tasks

Click on the "Snapshot" tabulator in the automation manager in order to edit your snapshot automation or to create a new task.

The snapshot tabulator

Automation	Active	Pattern	Query	Filename
Info	<input checked="" type="checkbox"/>	**** 0	/demo/abc.query	abc_snapshot.result
E-Mail	<input checked="" type="checkbox"/>	* 0 0	/demo/abc.query	

The table shows all snapshot tasks which are stored in your automation table. For each task you can edit the following options:

- **Active:** If this checkbox is selected, the Snapshot task is activated and will create snapshots with the time pattern defined in the Cron column.
- **Cron:** The Cron column controls, when and how often the snapshot will be created. To edit the cron pattern, you can use the edit button behind the pattern which will open the cron editor.
- **Query:** This mandatory field contains the path of the query you want to create snapshots for. The path must always be absolute to the root folder of the repository and must also contain the extension of the query (".query").
- **Format:** This field defines the format of the exported file. You can export a query as PDF, Excel or CSV file.
- **Target path:** If you define a filename or folder in this mandatory field, this will be the filename of the generated document or the target folder of the generated documents. If you provide a folder name here, the filename will be generated.

If you want to create a new export task, use the "Create task" button from the toolbar. This will add a new and empty task at the end of the task table.

Editing export tasks

Click on the "Export" tabulator in the automation manager in order to edit your export automation or to create a new task.

The export tabulator

Automation	Active	Pattern	Query	Format	Target Path
Info	<input checked="" type="checkbox"/>	* 0 0	/demo/abc.query	pdf	/home/reports

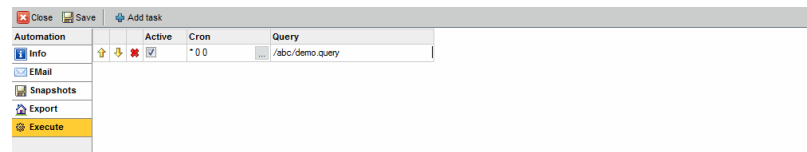
The table shows all export tasks which are stored in your automation table. For each task you can edit the following options:

- **Active:** If this checkbox is selected, the export task is activated and will create documents with the time pattern defined in the Cron column.
- **Cron:** The Cron column controls, when and how often the query will be exported. To edit the cron pattern, you can use the edit button behind the pattern which will open the cron editor.
- **Query:** This mandatory field contains the path of the query you want to create snapshots for. The path must always be absolute to the root folder of the repository and must also contain the extension of the query (".query").
- **Filename:** If you define a filename in this mandatory field, this will be the filename of the generated snapshot. The task will overwrite an existing file, so there will always be only one (the last version) of the snapshot. If no filename is configured, the task will automatically create a filename containing a timestamp.

Editing execute tasks

Click on the "Execute" tabulator in the automation manager in order to edit your execute automation or to create a new task.

The execute tabulator



The table shows all execute tasks which are stored in your automation table. For each task you can edit the following options:

- **Active:** If this checkbox is selected, the task is activated and will execute the query with the time pattern defined in the Cron column.
- **Cron:** The Cron column controls, when and how often the query will be executed. To edit the cron pattern, you can use the edit button behind the pattern which will open the cron editor.
- **Query:** This mandatory field contains the path of the query you want to execute. The path must always be absolute to the root folder of the repository and must also contain the extension of the query (".query").

If you want to create a new execute task, use the "Create task" button from the toolbar. This will add a new and empty task at the end of the task table.

Navigator

The navigator item of the tool list will open the navigator described before. Using this item has the same effect as using the "Show navigator" button from the repository toolbar.

Change password

Use this tool to change your password. After you clicked onto the item, a small dialog will open and ask you for your old and for the new password. You must confirm the new password in a second text field to ensure the new password is not misspelled.

The change password dialog

After entering the old and the new version, use the "Change password" button to change it. If anything is wrong, e.g. the old password or the confirmation, an error message will be displayed. Otherwise the dialog will be closed and your new password is now valid.

You can only see the "Change password" item in the tool list, if the internal user management of instantOLAP is activated. If your server uses an external user management, e.g. a naming service, instantOLAP will not be able to change a password and this item will not be visible.

CHAPTER 2:

Using the Workbench

Contents of this chapter:

System requirements	52
Workbench installation	53
Logging in	55
The desktop	57
Working with the repository	60
Manging folders	67
Tools	72

System requirements

For the Windows installation of the Workbench you need a computer with the following equipment:

- Microsoft Windows 2000, XP, Vista or Windows 7
- Intel Pentium IV / 2 GHz processor or faster
- At least 100 MB of free disk space
- At least 1024 MB of free memory

For the alternative Java Web-Start installation of the Workbench you need a computer with the following equipment:

- A Java Virtual Machine 1.5 or higher must be installed
- Intel Pentium IV / 2 GHz processor or better
- At least 100 MB of free disk space
- At least 1024 MB of free memory

Workbench installation

Installation variants

The instantOLAP Workbench is the remote administration tool for the instantOLAP server, which is executed locally on your workstation. Before you can use it, the software must be installed on your computer. There are two different ways to install the Workbench:

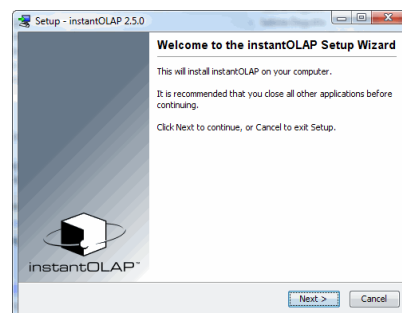
- Using the Windows-Installer
- Using Java Webstart

Installation using the Windows-Installer

This is the simplest way to install the Workbench: Download the newest version of the workbench from the instantOLAP website (<http://www.instantolap.com>) and choose the "Tools only" Windows-installer.

The "Tools only" package contains only the Workbench and no server components. It is delivered together with an integrated Java Virtual Machine and is a complete package - you don't need to install no further software on your computer.

The Windows installer



Start the executable installer and follow the dialog's instructions (you can use the directory and Start-Menu location). After the installer finished, the Workbench is ready to run. It can be started from the Windows-Start Menu (Programs/instantOLAP 2.5.x/Tools/Workbench).

Installation using Java Webstart

The Windows-Installer only works with Microsoft Windows. Also, the software will not be updated automatically every time if a new version is available. If you want to install the Workbench on another operation system than Windows or use the automatic update-feature for you clients, you should use the Java Webstart installation.

Java Webstart is an integrated installer for Java-programs which comes with the Java Virtual machine. It automatically downloads the software from the source location and checks for new versions every time you start the program.

To use the Webstart installer, the Java Virtual Machine must be installed on your computer before you can start the Workbench download.

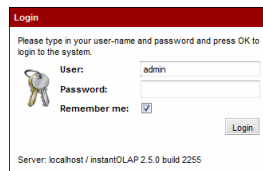
Install the Virtual Machine

To install the newest Virtual Machine, open the Java website <http://www.java.com> and navigate to the download of the software. The instantOLAP Workbench needs Java 1.5.0 or newer. Install the Virtual Machine before following the next steps.

Open the Web-Frontend and log in with your account

After you installed Java, start your browser and open the instantOLAP Web-Frontend. The URL of the frontend is "<http://<server-name>:<port>/iolap>" (the server-name and port depend on your instantOLAP installation). Then log in with your account.

The login dialog

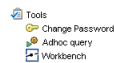


Remember that you need an account with administrator or Power User rights, otherwise you won't be able to follow the next steps.

Start the installation

After logging in, the Workbench item will appear under the tool explorer menu at left hand side of the portal.

The Workbench item



After using the button, the installer will open and start downloading the necessary files from the instantOLAP-server.

When the download has finished, a dialog will open and ask you to grant access to the computer for the software (because it may need to access your locale file system in future). Choose "Yes" and accept the software.

Now the Workbench starts and the login-dialog will become visible.

Starting the Workbench again

To start the Workbench later, simply click onto the workbench button again.

This will force the Webstart installer to compare the source files with your local installation and to start the Workbench immediately (without any download) if no file has changed. If a new version of the Workbench is available, the installer will automatically update your local copy of the software.

After you started the Workbench several time the installer will also ask you to create a desktop-icon for the software.

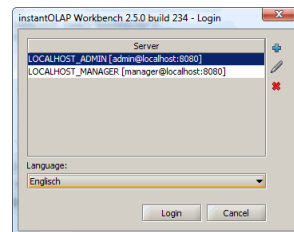
Installing the Webstart version from www.instantolap.com

You can also use the Webstart version offered on www.instantolap.com as installation source. Using this, you will always get the current version and the installer will automatically update whenever a new version is available.

Logging in

After you started the Workbench (either from the Windows Start menu or using the Workbench menu in the Web-Frontend), the login dialog appears with which you can log into any server you have added to the server-list before.

The login dialog



In the center of the dialog you can see the list of all instantOLAP servers which are already configured as targets for you Workbench. You can only log into servers which have been added before to this list.

The only server, which is visible directly after the installation of the Workbench, is the server "LOCALHOST" - the instantOLAP server, which is running on your computer locally. There are two different default settings for the localhost server, one for the user "admin" (which is the default administrator account), and for the user "manager" (which must initially be used to install a license on a server).

If no server is running on you computer (e.g. because you only installed the Workbench without server), or if you want to work on a remote server anyway, then you must add a new server-configuration now.

Below the server-list you can choose the language for the Workbench from the drop-down list. For the version 2.5 of instantOLAP, English, German and Turkish are available languages. By default, your Desktop language is selected (or English, if your language is not available).

Editing the server list

With the help of the buttons in the toolbar right beside the server list you can edit the list of the servers, i.e. add new servers and edit or delete existing entries.

Adding a new server to the server list

The "Add server" button



Use the "Add server" button to add a new server to the list. Then a dialog opens in which you can make the configuration settings for the new server.

Dialog for server properties

The following entries are available for each server:

- **Name:** The server name is the logical name, under which it will be displayed in the server list. You can assign any name for the server here and the name is mandatory.
- **Server:** The address or the IP of the server, followed by its port number. E.g., after a standard installation instantOLAP runs on port 8080, but any other port is possible.
- **Frontend:** This optional server address must be configured, if your instantOLAP frontend server is installed on a different machine than the backend, or if the frontend server has another context than "/iolap". Because the new version 2.5 of the Workbench uses the frontend server for its preview function, the Workbench has to know its address.
- **Proxy:** You can use optionally define a proxy server to be used to connect with the instantOLAP server. For this, type the proxy name or IP in the long field and the port of the proxy in the short field.
- **User:** The user account, under which you would like to connect the server. You must use a valid administrator account here (a user with the role "iolapAdmin").
- **Password:** Here you can type optionally the password of the user, so that this does not have to be queried with each login attempt. This is only for your comfort. Note that, for security reasons, it can be better not to provide a password since each user with access to your computer can connect the instantOLAP server without knowing the password.

After you made all inputs, you can use the button "OK" in order to store the new settings. If you use the button "Cancel" instead, all changes are lost and no server is added to the list.

Editing existing server entries

Existing entries in the server list can also be edited.

The edit server button



Use the "Edit settings" button to edit the server which has selected in the list before. Then the same dialog as when creating a server opens, but now with all fields being filled with the current settings of the server.

Change the configuration of the server as desired and then press the button "OK" in order to store the new settings. If you use the button "Cancel" instead, all changes are lost and the server remains unchanged.

Deleting a server from the list

The delete button



Select the server you would like to delete in the server list and use you then the button "Delete server" from the toolbar. After a safety inquiry the server will be deleted from the list.

Logging in

In order to log into a server, you must select it first from the list of the servers with the mouse. Then use the button "Login" at the lower edge of the dialog. Alternatively you can login by double clicking an entry.

If the server configuration does not contain a password, the password for the server will be requested. Otherwise this step is skipped you will see, if no errors occurs during the login, the desktop of the Workbench next.

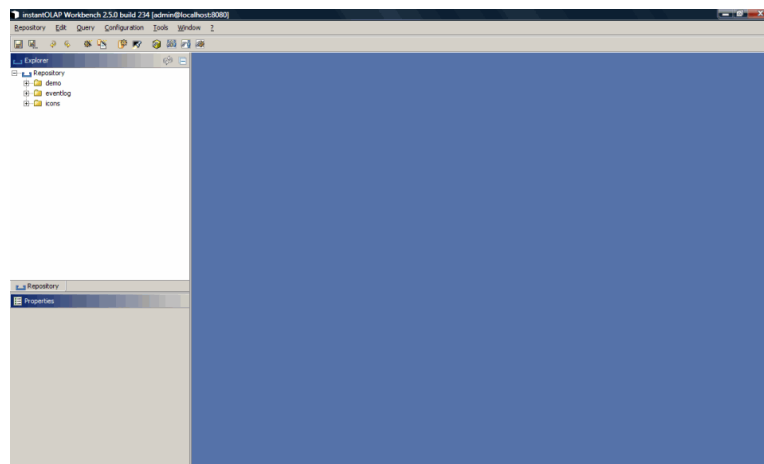
However if an error occurs during the login, e.g. because the server is not reachable, an error message will be displayed in the login dialog and you can try to login again. e.g. after changing the server configuration or solving the network problem.

The desktop

Elements on the Workbench Desktop

After logging in, the Workbench desktop opens. The Desktop is divided into three different main areas: The Repository Explorer (at the upper left) the Property Viewer (at the lower left) and the desktop area in the center, where all client windows open.

The desktop



Beside this areas, the Desktop also contains a menu-bar with all available functions and a toolbar for the frequently used functions.

The Workbench menu



With the menu, you can execute all functions of the Workbench. The menu is divided into different groups:



Repository	
Save	Save the contents of the current editor
Save As	Save the contents of the current editor with another file name
Save all	Save the contents of all editors
Refresh	Refresh the Repository-Explorer
Exit	Exit the Workbench
Edit	
Undo	Revert the last change
Redo	Revert the last undo
Query	











New query		Create a new query
Configuration		
New configuration		Create a new configuration
Import		
Import/From XMLA-Service		Create a configuration from a XMLA data source
Import/From Cubeviews	DB2	Create a configuration from a DB Cubeviews server
Tools		
Model query		Opens the Model-Query tool
Model manager		Opens the Model-Manager
Session manager		Opens the Session-Manager
User manager		Opens the User-Manager
Automation manager		Opens the Automation-Manager
Process manager		Opens the Process-Manager
License manager		Opens the License-Manager
Show/hide log		Shows or hides the logging panel
Window		
Arrange horizontally		Arranges all opened windows next to each other
Arrange vertically		Arranges all opened windows among themselves
?		
Info		Displays some system-information

The Toolbar



The toolbar offers most of the important operations from the menu as buttons. The following buttons are available:

	Save the contents of the current editor
	Saves the contents of the current editor with another file name

	Reverts the last change in the current editor
	Reverts the last undo
	Creates a new query
	Creates a new configuration
	Opens the Model-Query tool for querying a model directly
	Opens the logging panel on the desktop
	Opens the Model-Manager
	Opens the User-Manager
	Opens the Session-Manager
	Opens the Process-Manager

The Explorer

The Repository-Explorer in the upper, left corner of the Workbench is used for the file management, i.e. you can create, edit or delete folders, configuration, queries and other types of files here.

The Explorer



The Repository-Explorer also provides access to all important editors for the administration of folders, ie. for access-rights, automation, forwarding users etc. Most of this settings are made on the folder-level and you can open the appropriate editors using the context menu of a folder. You find a detailed description of the explorers and its functions in the section Working with the repository.

Working with the repository

The Repository-Explorer is used for the administration of folders and files and for editing several folder settings:

- The appearance of the folders (e.g. the titles, logos etc.) are edited with the Folder-Editor.
- The access-rights in instantOLAP are edited with the Access-Editor.
- Files and folders can be imported or exported with the context-menu of the repository.

Every of these editors is reachable over the Repository-Explorer with the context menu of a folder. Additionally, queries and model configurations can be created within a folder using the context menu.

Working with files

Creating files

How to create a file depends on its type: Configurations and queries can be created with the context menu of a folder or by using the toolbar of the Workbench (read the appropriate sections about the creation of queries or configurations). All other types of files, e.g. pictures, can only be created with an import into the repository (see below).

Renaming files

All files can be renamed, independently of the type of the file, using their context menu.

Use the context menu to rename a file



Open the context menu of a file with the right mouse button, while the mouse is over the appropriate file.

Open the menu item "Rename...". Then a small dialog appears, in which you can enter the new file name. Confirm your input thereafter with "Ok".

Note that changing the extension of a file can cause problems. E.g. if you change the extension of a configuration, whose name always with the extension ".config", then the system will no longer recognize the file as a configuration file. However, you can rename a file at any time again.

Copying files

There are two different methods to copy a file: Using the context menu of a file or with drag&drop.

Copying with drag&drop

Drag&drop is the most comfortable and simplest way of copying files within the repository.

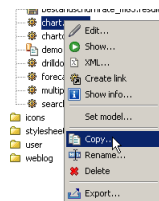
Select the file with the mouse and drag it, with pressed mouse button, into the folder, in which you would like to create the copy. Keep the "CTRL" key pressed when dragging the folder, otherwise the file will be moved instead of being copied.

With drag&drop you can only create copies in other folders than that, in which the file is located. If you want to create a copy of a file at the same place and with a different name, then you must use the context menu of the file.

Copying with the context menu

Use the menu item "Copy..." from the context menu of the file, in order to create a copy of the file in the same folder and with a different name.

Use the context menu to copy a file



After using the menu item, a small dialog appears, in which you are asked for the new name of the copy. Confirm your input with the "Ok" button and the copy of the file will be created.

Moving files

Like copying, moving of files within the repository is also possible with drag&drop. Select the file with the mouse and drag it, with pressed mouse button, to the target folder.

Do not press the "CTRL" key while dragging the file, otherwise the system will create a copy of the file instead of moving it.

Deleting files

Files can be deleted both with the keyboard and with their context menu in the repository.

Deleting with the keyboard

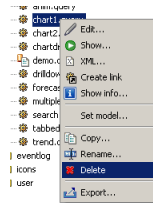
In order to delete a file with the keyboard, you must select it first, so that the background of the file appears dark.

Press the key "Delete" now in order to delete the file. Then a safety inquiry appears, which you must answer with "Yes".

Deleting with the context menu

For the deletion of a file you can also use its context menu.

Use the context menu to delete a file



Open the context menu and use the menu item "Delete" to delete the file. Now, like when deleting with the keyboard, a safety inquiry appears which you must answer with "Yes".

Exporting and importing files

With the Workbench, you can import and export individual files or complete folders from the local file system to the repository or vice versa. Read the section In- and export of files and folders for more information about the import and export.

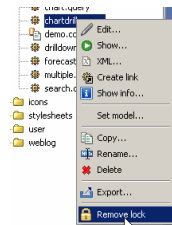
Unlocking files

When a user currently edits a file (e.g. a configuration or a query), the file will be locked until the user stops editing.

If you want to edit the file though it is locked, a message will point this out and the editor will not open until the file is unlocked.

You can unlock the file using the context menu. Use the menu item "Unlock" to delete the lock.

Use the context menu to unlock a file



Also, a file may still be locked if the user who locked the file did not close the Workbench in the regular way or if the Workbench stopped with an error. In this case, the Workbench will not have deleted the lock and you must do it manually.

Working with folders

Like in a file system, all queries and configurations are grouped and organized within folders. Folders can be nested in other folders in order to create a hierarchical organization of queries and configurations.

In the Repository-Explorer you can open and close folders, create new folders and rename or delete existing folders.

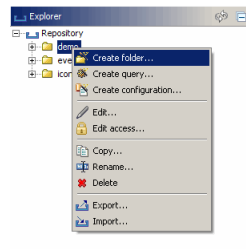
Opening and closing folders

In order to open a folder, you can click either on the small plus symbol before the listing name or double click on the name of the folder.

If the folder contains other folders or files, then these become visible after unfolding. The contents of a folder is only loaded from the server when opening the folder - therefore this can take some depending on the connection between the Workbench and the server.

Creating folders

You can create new folders using the context menu of the folder, under which you want to locate the new folder. If you want to create a new folder at the highest level of the repository then you must use the context menu of the repository node itself, which offers the same menu-item for creating folders.



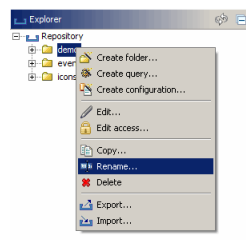
Open the context menu of the folder or repository with the right mouse button and use the menu item "Create new subfolder". Then a small dialog appears, which asks you for the name of the new folder. Enter the new name here and confirm your input with the "Ok" button.

The name of a folder is not necessarily the name, which the user can see later in Web-frontent, because you have the possibility to assign other names for the presentation in the Folder-Editor.

After creating the new and empty folder it appears below the containing folder in which you created it.

Renaming folders

Existing folders can be also renamed. Again, you must use the context menu to rename a folder.



Use the menu item "Rename..." of the context menu to rename a folder. Then a small dialog appears, in which you are asked for the new name. Enter the new name and confirm your input with the "OK" button.

Copying folders

There are two different way to copy folders within the repository: Using the context menu of the folder and using drag&drop. When copying a folder, all files and subfolders contained in folder are automatically copied, too.

Copying with Drag&Drop

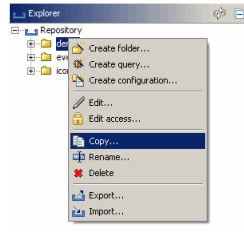
Drag&drop is the most comfortable and simplest way to copy folders.

Select the folder with the mouse and drag it, with pressed mouse button, to the folder in which you like to place the copy. Pay attention to it that you keep the CTRL key pressed when dragging the folder, otherwise it will be moved instead of being copied.

With drag&drop you can only create copies in other folders than that, in which the original folder is located. If you want to create a copy of a folder at the same place and with a different name, then you must use the context menu of the folder.

Copying using the context menu

The context menu of a folder also contains a menu item named "Copy...". Select this item, in order to create a copy of the folder in the same folder, but with a different name.



After the usage of the menu item a small dialog appears, in which you are asked for the new name of the copy.

Moving folders

Like copying, the moving of folders within the repository is also possible with drag&drop. Select the folder with the mouse and drag it, with pressed mouse button, to the target folder.

Do not press the "CTRL" key while dragging the folder, otherwise the system will create a copy of the folder instead of moving it.

Deleting folders

Folders can be deleted either with the keyboard or with their context menu. Note that all subfolders and files are also deleted with the deletion of a folder!

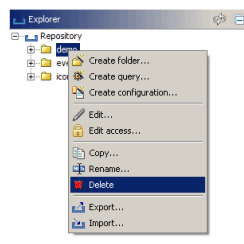
Deleting with the keyboard

In order to delete a folder with the keyboard, you must select it first, so that the background of the folder appears dark.

Press the key "Delete" now in order to delete the folder. Then a safety inquiry appears, which you must answer with "Yes".

Deleting using the context menu

For the deletion of a folder you can also use its context menu.



Open the context menu and use you the menu item "Delete" to delete the folder. Now, like when deleting with the keyboard, a safety inquiry appears which you must answer with "Yes"

Exporting and importing folders

With the Workbench, you can import and export individual files or complete folders from the local file-system to the repository or vice versa. Read the section In- and export of files and folders for more information about the import and export.

Import and export of files and folders

The im- and export of files and folders can be used to copy or move contents from the local file-system to the repositories of an instantOLAP-server or vice versa.

You can use the import and the export for different tasks when working with instantOLAP, e.g. for:

- copying configurations and queries from a development- to a production-server,
- the upload of images and other files onto the server, which are used by your queries,
- backup copies of the server files.

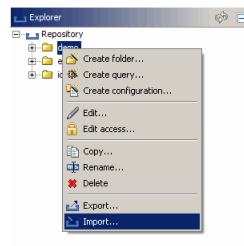
Importing files

Files and folders can be imported in different ways into the repository of the server: By using the context menu of a folder or by using drag&drop.

Import using the context menu

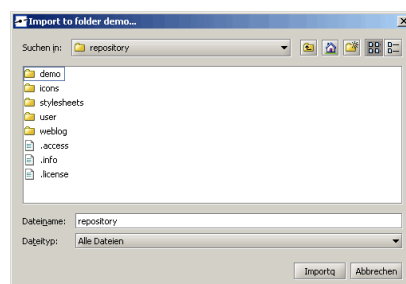
Use the menu item "Import" from the context menu of a folder to import files into this folder.

The import menu



After selecting the appropriate menu option, a file dialog appears, with which you can select a file or a folder from your local file system.

The import dialog



After the selection of the import source you must use the "Import" button - then all files are imported into the target folder. That can take, depending upon the number of files and the connection to the server, some seconds.

Import using drag&drop

Drag&drop is the simpler and more comfortable way to import files or folders into the repository of an instantOLAP-server.

Just drag files from the Windows-Explorer (or any other source) and drop them on the appropriate in the Repository Explorer.

While the import, you should hold the CTRL key, otherwise you will move the files from their original position.

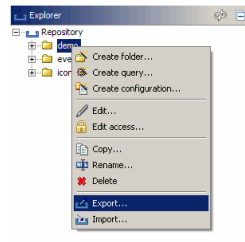
Exporting files

Files and folders can be exported with different methods from the Repository: Using the context menu of a file or folder and/or by using Drag&Drop.

Export using the context menu

Use the context menu of a file or a folder in order to export it to the local file system.

The export menu



Then a file dialog opens, in which you must select the target folder for the export. Use the button "export" to export the files.

Export using drag&drop

Like with the import, Drag&Drop offers a comfortable alternative to the context menus for the export. Drag the file or folder to be exported from the Repository into the Windows File-Explorer or any other target (e.g. the Windows-Desktop).

Pay attention to keep the CTRL key pressed in order to create a copy of the files and not to move them, otherwise they will be disappear from the Repository.

Manging folders

The administration of folders is a central task in instantOLAP. In addition to create, rename, delete or copy folder (described in the section Working with folders), there are several other tasks and editors for folders:

- With the **Folder-Editor** you can specify the title, stylesheets, logo and other properties for a folder. You can also make folders visible with this editor (new folders are initially not visible for the user of the system).
- With the **Access-Editor** you can specify, who may see and use the contents of a folder.

All editors can be reached by using the context menu of a folder. You can change at any time from one of these editors to the others.

The Info-Editor

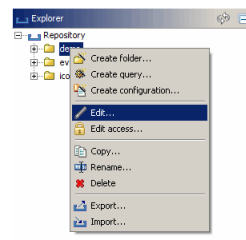
The Info-Editor allows to edit the basis properties of a folder. These properties cover:

- The visibility of a folder
- The title and the description of the folder
- The logos for a folder and its subfolders
- The stylesheets for a folder and its subfolders
- The sorting-style for the folder-content

Open the Info-Editor

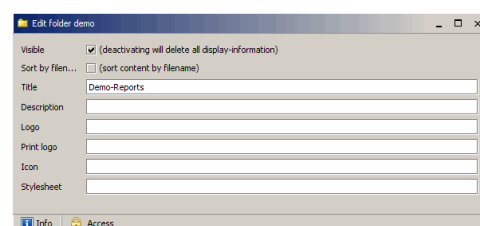
You must open the Info-Editor of a folder by using its context-menu. Select the folder and open this menu using the right mouse button.

The edit menu



Now select the menu option "Edit". Then the editor for this folder appears. Alternatively you can drag the folder on the Desktop area to open the editor.

The folder editor



Changing the folder properties

Visibility

In order to make an invisible folder visible, you must select the "Visible" checkbox. It will also automatically be activated as soon as you change any other, e.g. if you change the title. In order to make the folder invisible, you de-select the checkbox.

If you will make a folder invisibly, all fields in the editor will be deleted when saving the editor.

Sort by filename

Usually all files within a folder are sorted by their display-names (the name is a property which can be defined for each reports). By activating this switch, the files will be sorted by their filenames instead and you can manually define the order of the queries by changing their filenames (e.g. you may add numbers to each query-name).

Title and description

The title of a folder is the text being displayed in the Web-Frontend of instantOLAP. The actual name of a folder is not visible for a user.

The description of a folder is optional. It will be displayed in the index page of the Web-Frontend under the title of the folder.

Logo and print logo

The Logo and the Print logo are also optional properties. If you define logos here, they become the standard-logos for all queries in this folder and its sub-folder (both for Web and PDF-output). If no logo is defined, the standard-logo '/iolap/logo.gif' is used.

Icon

You can also change the icon of a query which is displayed in the index page of the Web-Frontend. Change this property to a valid URL to change the icon of the folder.

Stylesheet and CSS

With the "stylesheet" property you can change the standard STX-stylesheet of all queries in this folder or its subfolders. With the "CSS" property you can change the default CSS-stylesheets for the queries.

Saving the properties

The save button



You can save the changes of the properties by using the "Save" button from the Workbench toolbar. Alternatively you can use also the menu option "Repository/save" or the key CTRL-S to save the currently selected window.

If you made changes in the properties and close the editor without saving the changes before, the system will ask you to store the changes.

The Access-Editor

With the Access editor you can determine for a folder (and its subfolders), which user may see, execute or edit which report (or other files).

Function of the access control

The access control within instantOLAP is directory-oriented and works on the basis of rules.

Directory-oriented means that you configure the access to reports and other files in the directory in which the files lie. Should a directory have no access rules, then the rules from the parent directory are automatically used, so that you can also control the access for all underlying subfolders of a directory. Theoretically it is even possible to configure the rules for all files in the uppermost directory - the Repository.

Processing of the rules

Often there is more than one rule in a directory - then from the rules are checked from the first to the last, until a rule which applies on the current user, the desired operation and the affected file. According to the kind of the rule it permits (then it has the type "Grant") or forbids (then it has the type "Deny") the execution of the operation.

The order of the rules is very important. Should a rule apply, all following rules are no more considered by the system.

Matching of rules

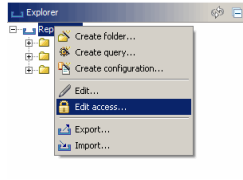
Whether a rule matches or not, depends on the entries in the columns "Role", "User", "Action" and "Entry". In these columns the user names, groups, operations and entries (filenames) are configured to which a rule should apply (unless you use the wildcard "*" in any of the columns, which then matches for any user, group etc).

- **User:** The name of the user who would like to perform an operation. Should the name of the user match with this field or the wildcard "*" is used here, then this column matches.
- **Role:** The name of an user-group (role). Should the current user own this role or "*" is used as a wildcard, then this column matches.
- **Action:** There are two access levels: "Execute", and "Admin". Also, the wildcard "*" can be used (which has the same meaning as "admin").
- **Entry:** This is a pattern for the name of the files to which this rule applies. You may also use the Wildcard "*" here - however, in difference to the other columns the wildcard can also match only parts of the name. E.g. a pattern "demo*" matches for all files which begin with "demo". A pattern "*.config" matches for all configurations etc.

Should you also want to configure the rules for files from its subdirectories in a directory, you must put the directory name of the subdirectory (followed by a slash "/") before the filename. E.g. a pattern "demo/" matches for all files from a subdirectory named "demo".*

Opening the Access-Editor

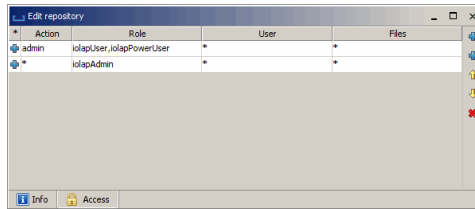
Use the context menu of a directory to configure the access rules for a directory.



Open the context menu of the directory with the right mouse button and use you the menu item "Edit access.." This opens the Access editor.

Alternatively you can also open the directory editor and switch to the Access editor using the tabulator "Access". You also reach the directory editor with the context menu (using the menu item "Edit ...") or by dragging the directory on the surface of the Workbench using drag&drop.

The access editor



Editing the entries

You can create new entries, move entries or delete entries. They with the tool bar at the right margin of the editor.

Editing entries

To change the values of an entry, you must click with the mouse in a column within the table. Now you can change the value and confirm the update with the Enter key.

Creating new entries



Use the button "Add new rule" to add a new rule to the list. Then an empty rule is inserted at the end of the list which you can edit now.

Moving the entries



With the arrows from the tool bar you can change the order of the entries.

Choose the entries which you would like to move and press the suitable button. Then the entries are moved one position in the list upwards or down.

Deleting entries

Delete button



Use the button "Delete rule" to delete the currently selected rules from the list.

Saving the settings

Save button



You can save the changed settings any time using the save button from the tool bar of the Workbench.

Alternatively you can also use the menu item "Save" or use the keyboard shortcut CTRL-S if the access editor is the currently selected window.

If you made changes in the configuration and then close the editor without having saved before, then the Workbench will point out you to these changes and ask whether they should be saved.

Tools

For the administration instantOLAP-servers a set of tools exist, with which you can perform different tasks:

- **User-Manager:** With the User-Manager you can create users and user groups as well as assign users to user groups.
- **Automation-Manager:** The Automation-Manager allows to set up the backend automation for the current user or for any user if the current user is an administrator. The backend automation includes tasks like sending queries as E-mails, exporting documents etc.
- **Model-Manager:** The Model-Manager shows all existing models on the server. With the manager, models can be started, stopped or rebuilt.
- **Process-Manager:** The Process-Manager shows all actively running query or other process. Current processes can also be canceled with this tool.
- **Session-Manager:** The Session-Manager shows all active user sessions on the instantOLAP-server. You can also cancel sessions with this tool.
- **License-Manager:** With the License-Manager you can administer the installed licenses on a instantOLAP-server and add and/or delete new licenses.
- **Logging panel:** The logging panel allows to view internal processes of a instantOLAP-server. E.g. the exact operational sequence of an query can be analyzed with the logging panel, i.e. for error tracing or performance analysis.

User-Manager

The User-Manager allows the administration (the creation or deletion) of user-accounts, -groups and roles. Also, individual users or whole user-groups can be assigned to the roles which are used by the access-control of instantOLAP.

The User-Manager is only available if the instantOLAP server uses its own user management. Depending on your installation it could also be possible that an external user-database is used for your instantOLAP-server, which then cannot be administered with the Workbench (e.g. a Windows Naming server or a LDAP database).

Opening the User-Manager

The user manager can be opened using the menu "Tools/User manager" of the Workbench or with the button "User manager" in the toolbar.



After pressing the buttons or using the menu, the window of the User-manager opens. The window is divided into two main areas, the user-editor and the role-editor. You can reach both editors with the appropriate tab at the lower edge of the window.

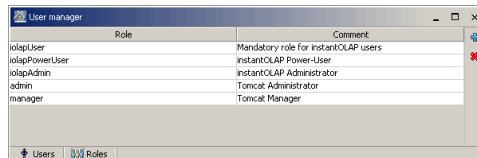
Roles

In instantOLAP, like in all Java Web applications, users are assigned to "roles" in order to simplify the administration of the access-control. All access-rules are defined on the basis of roles and therefore all users, who own a certain role, will be able to perform all actions which a granted for a certain role.

E.g. if assign a user to a role later, he receives automatically all rights of access, which are defined for this role.

Opening the role editor

If you don't see the role-editor currently, you can switch to it with the tab "Roles" at the lower edge of the window.



Creating new roles



Use the button "Add role" to add a new role to the list of all existing roles. Then a dialog opens in which you can enter the name of the new role. The name of the role can be changed afterwards.

Editing roles

You can edit new or existing roles by clicking in a cell within the table and edit its values directly. There are following values for roles which can be edited:

- **Name:** The name of a role
- **Description:** An optional description of a role which is visible only in this editor

All changes in a role become effective immediately - you don't need to save them.

Deleting roles



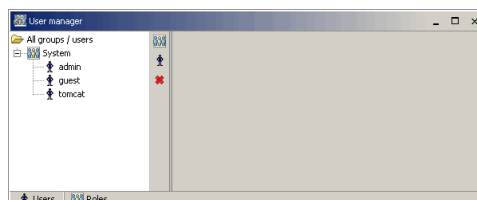
In order to delete an existing role select it first and then use the button "Delete role".

Groups

Groups are used to group users (or other groups) in order to make them easier to administrate. Also, you can assign roles to groups and automatically make all users of a group to members of this roles.

Opening the group editor

Groups can be edited in the user editor of the manager. If you don't see the user editor in the manager, you can switch to it with the tabulator of "user" in the lower edge of the manager.



Creating new groups



You can create a new user group with the "Add group" button. After pressing the button a new group will appear under the currently selected group in the tree representation at the left side of the user editor.

If no group was selected, the new group will be added under the root element of the tree. If a user was selected, the new group will be added to the group the user belongs to.

At the right side of the editor, the new group with all of its properties will be displayed now. You can edit the following properties:

- **Name:** The name of the new group
- **Comment:** This optionally comment will only be visible inside the User-Manager

Assigning roles to groups

You can also assign roles to groups. All members of this group or any subgroup of this group will automatically inherit the roles being assigned to this group.

To assign one more roles to a group you must simply activate the corresponding checkbox in the list of roles below the group properties. To delete an assignment you must deselect the checkbox of a role.

Saving groups



Use the "Save" button in the tool bar at the right side of the group editor to save the currently edited group.

Editing existing groups

You can also edit an existing group by selecting it in the tree view at the left side of the user manager. After selecting the group, its properties and the assigned rules will be displayed at the right side of the manager.

Don't forget to save the group after changing its properties, otherwise your changes will be lost after closing the manager or displaying another group or user.

Deleting groups



Use the "Delete" button in the tool bar next to the tree view in order to delete a group after you selected it in the tree.

Editing users

Opening the user editor

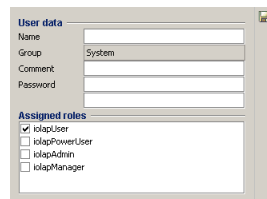
Like groups, users can be edited in the user editor of the manager. If you don't see the user editor in the manager, you can switch to it with the tabulator of "user" in the lower edge of the manager.

Creating new users



You can create a new user with the "Add user" button. After pressing the button a new user will appear under the currently selected group in the tree representation at the left side of the user editor.

If no group was selected, the new user will be added under the root element of the tree. If another user was selected, the new user will be added to the group the selected user belongs to.

A screenshot of the 'User data' editor window. It has a title bar 'User data' and a close button. The window is divided into two main sections. The top section is labeled 'User data' and contains four input fields: 'Name' (empty), 'Group' (containing 'System'), 'Comment' (empty), and 'Password' (empty). The bottom section is labeled 'Assigned roles' and contains a list of roles with checkboxes: 'ldapUser' (checked), 'ldapPowerUser' (unchecked), 'ldapAdmin' (unchecked), and 'ldapManager' (unchecked).

At the right side of the editor, the new user with all of its properties will be displayed now. You can edit the following properties:

- **Name:** The name of the new group
- **Comment:** This optionally comment will only be visible inside the User-Manager
- **Password:** The password of the user, displayed in two input fields. If you change the password, you must retype it in the second field to confirm it.

Assigning roles to users

To assign one more roles to a user you must simply activate the corresponding checkbox in the list of roles below the user properties. To delete an assignment you must deselect the checkbox of a role.

Saving to users



Use the "Save" button in the tool bar at the right side of the user editor to save the currently edited user.

Editing existing users

You can also edit an existing user by selecting him in the tree view at the left side of the user manager. After selecting the user, his properties and the assigned rules will be displayed at the right side of the manager.

Don't forget to save the user after changing its properties, otherwise your changes will be lost after closing the manager or displaying another group or user.

Deleting users



Use the "Delete" button in the tool bar next to the tree view in order to delete a user after you selected him in the tree.

Automation-Manager

Automation in instantOLAP allows to send EMail, export queries or to perform other tasks in the background.

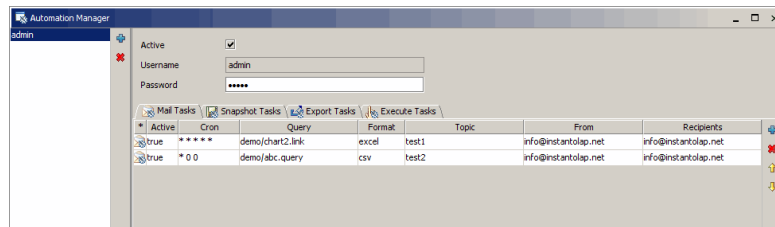
Each user has its own automation table where all his automations are stored. In difference to previous versions of instantOLAP, the automation is no longer folder based but user based. This is because since version 2.5 users are able to create and configure automation on their own.

Depending on your roles, you can use the Automation-Manager to manage you own automations or, if you own the administrator role, to manage the automation of all users, too.

Opening the Automation-Manager

To open the manager, you must use the menu item "Tools/Automation-Manager" of the Workbench menu. After that, the window of the License managers opens.

The Automation-Manager



At the left border of the window, the list of all users owning an automation table are shown in a list. Only administrators can see this list, otherwise the manager will only show the automation the current user and offer no user list.

The panel on the right side shows all existing automations for the selected or current user. Because there are different types of automations, they are organized in a tab panel, with one tabulator per automation type:

- **Email tasks:** This tabulator contains all EMail tasks for the current user. EMail task allow to export queries, pivot-tables or links and to send them via EMail.
- **Snapshot tasks:** Snapshot tasks execute queries and save the result as snapshots in the same folder the query is located.
- **Export tasks:** These task allows to execute queries and to export the result as document (PDF, Excel or CSV) onto the file system of the folder.
- **Execute tasks:** Execute tasks simply execute queries. Whenever a query is executed, the caches of the model will be filled with the new values. Using this task can result into a better performance, when human users execute queries.

Only administrators can see all of these tabulators, all other users can only see the EMail automation.

Above the task tabulators, some general informations are displayed:

- The **active** flag indicates, if the complete automation for the user is enabled or disabled.

- The **user** field show the name of the user.
- The **password** field must contain the current password for the user. Because the backend server will start sessions for this user, the automation table must also know his password.

The user list

The user list at the left side of the manager is only visible for administrators and is used to create or delete automation for users.

Creating new automations

Add user



To add a new user to the list, use the "Add new user" button from the toolbar right to the list. A dialog will open and ask you for the name of the user.

The username you enter here must exist, because it does not add a user to the user database - It only creates an automation for an existing user.

After you entered the name, an empty automation opens on the right half of the automation manager. The user name is already pre-configured with the name of the user, but you'll have to enter the correct password for the the user here.

Usually an administrator does not know the name of a user. But for general automations it is recommended to create a technical user (e.g. "automation") and to configure all general automations for this user.

Now you can edit the automations of the user at the right side. Note that the automation will not be stored on the system unless you use the "Save" button from the Workbench toolbar.

Deleting an automation

Delete user



If you want to delete the complete automation for a user, select his username in the user list and use the "Delete user" button from the toolbar at the right border of the lust.

After you confirmed the following dialog, the automation for the user will be deleted. Note that the deletion is already performed at this point, even if you don't use the "Save" button afterwards.

Editing automations

The right half of the automation manager shows all automation of the current user or the selected user in a tabulator panel. Click on one of the tabulators to display and edit the tasks:

Editing EMail automations

Click onto the "EMail" tabulator to edit the EMail automations.

EMail automations

* Active	Cron	Query	Format	Topic	From	Recipients
true	****	demo/chart2.link	excel	Chart example	info@instantolap.net	info@instantolap.net
true	** * 0,12 0	/demo/abc	pdf		info@instantolap.net	test@instantolap.net

All EMail tasks are display in a table with the following columns:

- **Active:** This flag indicates whether the automation is active or not. If you deselect the checkbox, this single automation will not be executed in the future.
- **Cron:** The cron pattern determines, when and how often the automation will be executed. Use the edit button "..." right to the pattern to edit it in a more comfortable editor.
- **Query:** This field contains the absolute path of the query, link or pivot-table which should be executed and send as mail, including the extension of the file.
- **Format:** The EMail task can send the query result as PDF, Excel or CSV document. Choose the format with this dropdown box.
- **Topic:** This is the topic the generated EMail will contain. If you provide no topic, the system will use the query name as default topic.
- **From:** You can enter the EMail address which appears as sender in the mail here. If you leave the field empty, the default sender of the EMail server is used.
- **Recipients:** List all recipients with their EMail addresses here. If you want to send the mail to more than one recipients, you must concatenate their addresses with comma ",".

If you want to edit the existing entries, simply click into the table cells and edit their values. Only the cron pattern has an additional cron pattern editor which can be reached by clicking onto the edit button.

Add task



Use the "Add task" button to add a new row to the list. The new task will be initially empty and can be edited inside the table.

Delete tasks



Use the "Delete selected tasks" button to delete a row after you selected it by clicking on the image in the first column. The button will delete the row from the table without any confirmation.

Move tasks



If you want to change the order of the tasks you can select one or more tasks and use the button "Move task up" or "Move tasks down" to shift task within the list.

Editing Snapshots automations

Click onto the "Snapshot" tabulator to edit the snapshot automations.

Snapshots automations

* Active	Pattern	Query	Filename
<input checked="" type="checkbox"/>	*** 0 0	/demo/abc.query	
<input checked="" type="checkbox"/>	*** 0 0	/demo/trend.query	trend.result

All snapshot tasks are display in a table with the following columns:

- **Active:** This flag indicates whether the automation is active or not. If you deselect the checkbox, this single automation will not be executed in the future.
- **Cron:** The cron pattern determines, when and how often the automation will be executed. Use the edit button "..." right to the pattern to edit it in a more comfortable editor.
- **Query:** This field contains the absolute path of the query, including the extension of the file.
- **Filename:** This optionally field contains the filename of the result. If you leave the field empty, the system will create a new file every time and use a generated filename containing a timestamp. If you provide a filename (which must be relative to the folder and end with the extension ".result"), the system will always overwrite the result.

The task can be edited, created, deleted or rearranged like the "EMail" tasks described before.

Editing Export automations

Click onto the "Export" tabulator to edit the export automations.

Export automations

Active	Pattern	Query	Format	Target Path
<input checked="" type="checkbox"/>	* * * * 0	/demo/abc.query	pdf	/home/olap-exports

All snapshot tasks are display in a table with the following columns:

- **Active:** This flag indicates whether the automation is active or not. If you deselect the checkbox, this single automation will not be executed in the future.
- **Cron:** The cron pattern determines, when and how often the automation will be executed. Use the edit button "..." right to the pattern to edit it in a more comfortable editor.
- **Query:** This field contains the absolute path of the query, including the extension of the file.
- **Format:** This dropdown lets you choose the format for the exported files. Possible formats are "PDF", "Excel" or "CSV".
- **Target Path:** This field contains the folder or the filename of the result. If you provide a folder, the system will create an automatic filename containing a timestamp and save the result in the folder. If you provide a complete filename, the system will use it and overwrite existing files.

The task can be edited, created, deleted or rearranged like the "EMail" tasks described before.

Editing Execute automations

Click onto the "Execute" tabulator to edit the execute automations.

EMail automations

Active	Cron	Query
<input checked="" type="checkbox"/>	* * * * 0	/demo/abc.query

All snapshot tasks are display in a table with the following columns:

- **Active:** This flag indicates whether the automation is active or not. If you deselect the checkbox, this single automation will not be executed in the future.
- **Cron:** The cron pattern determines, when and how often the automation will be executed. Use the edit button "..." right to the pattern to edit it in a more comfortable editor.
- **Query:** This field contains the absolute path of the query, including the extension of the file.

The task can be edited, created, deleted or rearranged like the "EMail" tasks described before.

Saving

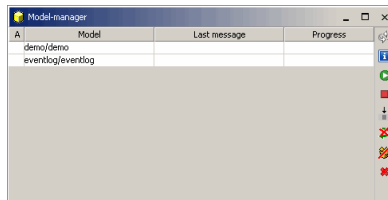
All changes you make in the automation for user will not be stored on the server unless you use the "Save" button from the Workbench (or the menu).

Model-Manager

The Model-Manager is the central tool for the control of all models running on the instantOLAP server. It can be used to start and stop models, to rebuild their dimensions and stores or to clear the caches and persistent data of the models.

Opening the Model-Manager

Use the menu-item "Tools/Model-manager" or the Model-Manager button from the Workbench toolbar to open the manager.



Working with the Model-Manager

Informations about models

For each model there are three different informations being displayed as columns in the table of the manager:

- **State:** The first column of the table shows the state of a model. There are four different possible states: Stopped (no sign), Running, Paused/Offline and synchronizing.
- **Model:** The full name of the model, including its full path. Temporary models (which are created when editing a configuration) are also displayed here, they contain the character '~' in their name.
- **Last message:** If the start or the synchronization of a model failed, the reason is displayed in this column.
- **Progress:** Shows the progress of a model if it is currently starting.

You can open the model-info window for a single model to see more detailed information.

Refreshing the list



Use the button "Refresh" to refresh the list of the models and their states.

Opening the Model-Info for a model



Use the button "Show model info" to open a window with more detailed information for the currently selected models. Alternatively you can also double-click onto the model inside the table. See below for a description of the Model-Info window.

Starting models



With the button "Start selected models" you can start all models being selected in the model-table.

After you started a model, the progress of the startup will be displayed as a bar (with percentage) in the column "Progress" of the table. If an error occurred while starting a model, the last error-message will be displayed inside the table. Otherwise, the model will switch into the "Running" state after it successfully started.

Stopping models



Use the button "Stop selected models" to stop all selected models.

Rebuilding models



With the button "Rebuild selected models" you can rebuild all dimensions of all selected models. Rebuilding dimensions means to reload all elements of a dimensions.

Clearing caches



The button "Clear caches of selected models" will clear all caches. You can also use the cache-tabulator from the Model-Info to clear single caches from the model.

Clearing stores



Use the button "Clear stores of selected models" to empty all stores of the selected models. If the models are configured to auto-rebuild, the system will start filling the stored within the next minute. You can also use the store-tabulator from the model-info to clear single stores of a model.

Deleting the persistent data of a model



The button "Delete persistent data of selected models" will delete all persistent data of a model. This includes the persistent dimensions, caches and offline cubes of a model. Note that a model will automatically be stopped before the systems deletes its data.

Working with the Model-Info dialog

Opening the Model-Info for a model

To open the model-info you can double-click onto a model or use the button "Show model-info" from the toolbar of the Model-Manager.

Viewing dimensions

The first tabulator of the model-info displays all dimensions of the model with several informations:

- The name of the dimension

- The current status of the dimension. A dimension can be "loaded", "synchronizing" or "rebuilding"
- The number of keys stored in the dimension
- The timestamp of the last actualization. This is the timestamp of the last successful rebuild of the dimension.

The dimension panel

Name	State	Keys	Last Rebuild
Fact	Online	6	Oct 29, 2008 1:02:52 PM
Product	Online	104	Oct 30, 2008 3:06:06 PM
Manufacturer	Online	9	Oct 30, 2008 3:06:07 PM
Campaigns	Online	6	Aug 14, 2008 3:21:31 PM
Time	Online	27	Aug 14, 2008 3:21:31 PM
Agency	Online	135	Aug 14, 2008 3:21:31 PM
Anm	Online	101	Aug 14, 2008 3:21:31 PM
LINE_DIMENSION	Online	1	Oct 30, 2008 3:36:13 PM

At the right border of the panel there is a toolbar with several buttons:

Refresh



The "Refresh" button refreshes the list of dimensions and updates their current state informations.

Rebuild selected dimensions



This button rebuilds all selected dimensions and all other dimensions which are dependent on them. A dimension can depend on another if both are linked with a relink-attribute. This makes it necessary to reload both dimension.

Viewing cubes

The cube panel displays all cubes of a model and their current state:

- The first columns show the "Name" of the cube
- The second column shows the current state of a cube. The state can be "Adhoc", "Adhoc + loading", "Offline" or "Offline + loading". "Adhoc" means, the cube takes the data from the SQL database with generated SQL queries, "Offline" means the cubes uses the data from its preloaded store. If the state also contains "+ loading", the cube is currently loading its data and the progress columns display the current progress.
- The "Size" column shows the number of loaded entries for a offline cubes. This is only the number of entries containing data, not the number of possible combinations of its dimensions, which normally would be much larger.
- The "Last actualization" shows the timestamp of the last successful loading of an offline cube.
- The "Progress" column shows the progress of a currently loading cube. This is the percentage of processed input rows. Because of the loading algorithm, the last 10 percent usually take more time to load than the first 90.
- The "Load time" shows the duration of the last successful offline load in seconds.
- The "Last error message" show any error which occurred in the last offline load. If an error is shown here, the last load was not successful and the data in the cube is not up to date.

The cube panel

Name	State	Size	Last rebuild	Progress	Build time (s)	Error message
SALES	Loaded	81595	Oct 30, 2008 ...		3.047	

You can use the toolbar at the right border to load or delete the offline data of the cubes:

Refresh



The "Refresh" button refreshes the list of cubes and their states.

Rebuild selected cubes



This button reloads the selected cubes and tries to load their data into its offline store. Only cubes with names can be rebuild, and only cubes with mappings configured for offline loading will contain data after loading them.

Clear selected cubes



This button deletes the offline data of the selected cubes and sets them back to the "Adhoc" state.

Viewing caches

The last tabulator displays information about all caches of the model. For each existing cache, different informations are displayed:

- The name of the cache
- The size (the number of stored entries) of the cache
- The timestamp of the last cleaning of the cache. A cleaning deletes all outdated entries from the cache.

The cache panel

Name	Size	Last check
CACHE1	10916	Oct. 30, 2008 3:34:57 PM

At the right border of the panel there is a small toolbar with two buttons:

Refresh



The refresh buttons refreshes the cache list and display their current state.

Empty selected caches



The "Empty cache" deletes all entries from the selected caches.

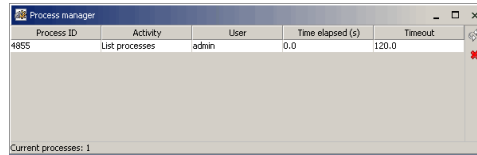
Process-Manager

The process manager serves for the monitoring and control of all actually running queries, reports and other processes on an instantOLAP server.

Opening the process manager

You can open the process manager, like the other tools, either by using the menu of the Workbench or with the tool bar of the desktop. Use the menu item "Tools/Processes-Manager " or the button "Process-Manager" from the tool bar.

The process manager



In the manager all actually running queries are shown as a single row in a list. For every query you can see the following information:

- **ProcessID:** The ID of the running process (every execution of a query has its own ID)
- **Query:** The executed query
- **User:** The user who executes the query
- **Time elapsed (s):** Number of seconds which have passed since the start of the query

Actualization of the view



With the "Refresh" button of the Process-Manager you can update the view of the processes any time.

Stopping Processes

Use the "Kill process" button to terminate running processes: First Select the processes in the list which you would like to kill and then press the button.

A user will receive a error message in the Web-Frontend if a his report was interrupted the Process-Manager.

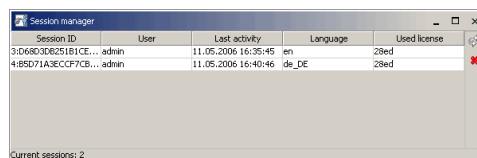
Session-Manager

With the Session-Manager you can supervise all active sessions on an instantOLAP server and also kill them if necessary.

Opening the Session-Manager

Use the menu of the Workbench (the menu item "Tools/Session-Manager") or the button "Session-Manager" from the tool bar for opening the Session-Manager.

The session manager



In the Session-Manager all actual sessions which run on the instantOLAP server (both for the users of the web frontend as well as of the Workbench) are shown.

For every session a number of information is displayed:

- **ID:** The unique ID of the session
- **User:** The user account to which the session belongs
- **Last activity:** The time of the last user activity

- **Language:** The language of the session
- **Used License:** The number of the used license

In the license number only the first four digits of the license code (and not the complete code) are shown. With the help of the license number you can see, to which "Named License" a user is assigned.

Refreshing the list

Refresh



With the "Refresh" button from the tool bar at the right edge you can refresh the view of the Sessions any time.

Killing a session

Kill session



To kill a session, you must select this before and then use the "Kill session" button from the tool bar.

You can kill all running sessions except your own on the server. If you try to kill your own session, you will receive an error message.

License-Manager

The License manager is used to install new licenses on the instantOLAP server, to delete existing licenses and to associate certain user accounts to a "Named License".

Opening the License manager

To open the manager, you must use the menu item "Tools/License-Manager" of the Workbench menu. After that, the window of the license managers opens.

The License-Manager

License No	User type	License type	Size	Expires	Assigned users
DC74	Admin	Concurrent	3	04.06.2006 12:2:2...	

In the License-Manager all licenses which are installed currently on the server are displayed in a list. The following information is displayed in each row:

- **The license number:** Every license owns an unique license number which is displayed here.
- **The user type:** A license key permits either sessions for an Administrator, a Power-User or Thin-user. Better licenses are also used for logging on "smaller" user-types (what only happens if no more enough "small" licenses are available).
- **The license type:** Licenses can be either "Named Licenses" or "Concurrent Licenses". "Named Licenses" are associated to certain users - these users have a guarantee that they can log in at any time, because they use a dedicated license. "Concurrent Licenses" are shared between all other users (who are not assigned to any "Named License"). These user may not be able to login if the amount of available "Concurrent Licenses" is used up.
- **Size of the license:** The number of the users who can use this license key at the same time. In case of "Named License" that is the number of the users who can be assigned to

this license. In the case of "Concurrent License" it its the number of the users without "Named License" which can log into the system at the same time in.

- **Expiration of the license:** Licenses can have a limited lifetime. In this case, the date of the license's expiration is displayed here. Otherwise the text "Never" is displayed.
- **Assigned users of a Named-License:** Single users can be assigned to a "Named License" - then in this column the names of the users which belong to a "Named License" appear.

Installation of the first license

With to the first connection with the Workbench to an a newly installed server without no installed license the License-Manager automatically opens and you can install the first license before you work with the server.

Adding a new license

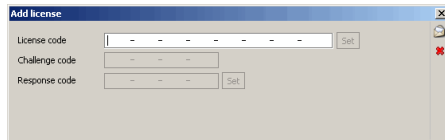
With the "Add license" button from the tool bar in the right margin you can add other licenses to your server.

Add license



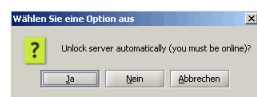
After using the button another dialog opens in which you can enter the code of the new license and unlock it:

The license dialog



1. At first enter the license code in the first field and confirm your input with the button "Set" on the right beside the field.

If the license code is valid and no error was found in the input, the license must be unlocked in the second step. To "unlock" means that you register the use of the license on the instantOLAP web server.



For the unlock of a license you'll need a "Challenge code" which contains the number of the license and the used IP-address of the server - this "Challenge code" is displayed in the second line of the dialog after the input of the license.

2. With this "Challenge Code" you can create a "Response Code" on the web page of instantOLAP (www.instantOLAP.net) which you must enter then in this dialog. Use the button "Set" beside the field "Response Code" afterwards to activate the license.

You can also allow let the dialog perform the automatic activation of a license in order to skip the manual creation of the Response Code. This will transmit only the Challenge Code and Response Code between the instantOLAP server and your workstation. You are asked by the system after the input of the License Code whether an automatically activation should be performed. For this connection your computer must be online and the Workbench must be able to connect the server www.instantOLAP.net.

Deleting an installed license

Delete



Select the license which you would like to delete in the list and use the button "Delete license". Then, after a query, the license is deleted from the license pool.

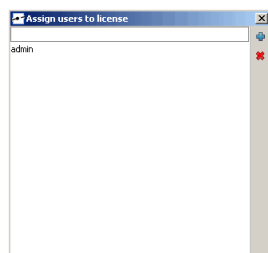
If you delete the administrator-license under which you work currently, a new dialog appears immediately which requests you to the input of a new administrator-license. If you enter no new license here, your current session will be killed.

Assigning users to a "Named License"

If you have installed one or more "Named License" on you system, they are not used until you have assigned at least one user to these licenses.

To assign an user to a Named License you must select the license and then use the "Assign Users" button from the tool bar.

Assign user dialog



Now another dialog, in which all users assigned to the license up to now are displayed in a list, opens. Above the list a small input field, in which you can enter the name of a new user which you would like to assign, is visible.

Adding users

In order to assign another user to a license you must first enter the name of the user in the text field above the list.

Assign user



The use the button "Assign user" to assign the user and to add him to the lower list. Should the user be already assigned to this license, this operation is ignored.

If an user has been already assigned to another license, the previous assignment gets lost.

Removing users

Remove user



With the button "Delete selected assignments" you can remove the users, which you have selected in the list before, from the license . Only the assignments are deleted by this - not the users themselves.

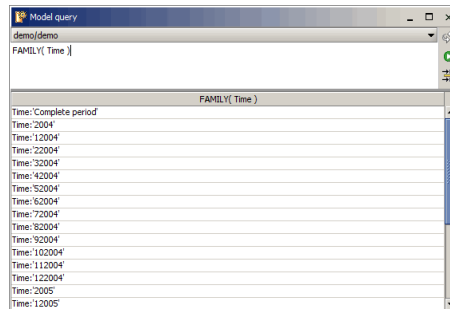
The Model-Query tool

With the Model-Query tool you can perform adhoc queries in the form of expressions on models. This tool serves primarily for the testing and exploration to know of models. With the Model-Query you can perform *no* graphic report on models.

Opening the Model-Query

Use the button "Model-Query" from the tool bar of the Workbench or the menu item "Tools/Models-Query". Then the window of the tool appears.

The model query tool



In the window different elements are shown:

- The large input field is the place where you can type the expression of the query you want to execute
- The dropdown box above the input field is the model-selector with which you can select the model you want to execute the query on.
- In the toolbar at the right border of the field there are three buttons. With the refresh-button you can refresh the model list, with the execute-button you can execute your query and with the limit-button you can set the maximum number of result-rows.

In the table below the input field the results of your queries will become visible. This table is empty after opening the tool for the first time.

Selecting a model

First select the model on which you would like to perform a query. A list of all available models is shown in the Drop-Down above the input-field.

Executing to a query

The execute button



Type your query into the text field "Query". To execute this query on the actually selected model, you must use the button "Execute query" right beside the field. As an alternative, you can also press the keys CTRL+ENTER.

If there is a selection inside the input-field, the model-query will only execute the selected part of the text. Otherwise, the whole expression will be executed.

The result of your query is shown after the successful execution in the table below the input field. In the case of an error, e.g. if the syntax of the expression was not correct, an error message is shown instead of the result.

Refreshing the tool

Refresh



With the button "Refresh" in the upper, right corners you can refresh the list of the models in the dropdown-box. If new models have been added to the system, they will become visible then.

Changing the row limit

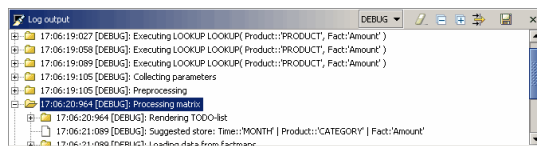
By default, the result will be limited to the first 1000 rows. You can change this limitation with the button "Change maximum result size" from the toolbar.

The Logging-Panel

With the Logging tool you can understand and examine the execution of queries and the build of models. The Logging tool is a docking-panel which is shown in the lower edge of the Workbench.

Opening the Logging-Panel

To open the Logging-Panel you can either use the button "Show Log" from the tool bar of the Workbench or the use menu item "Tools/Show Log".



Within the tool all events are shown in a growing tree-list. This list is hierarchical and nested entries are encapsulated into each other and can be folded or unfolded within the tree.

Adjusting the log level

The most important control for the Panel is the adjustment of the log level, with which you can determine how detailed the messages will be. The following log levels are available:

- **NONE:** No messages are generated
- **ERROR:** Only error messages are shown
- **WARN:** Error messages and warnings are shown
- **INFO:** Beside the error messages and warnings all of your activities are reported, e.g the execution of a report
- **DEBUG:** Like INFO, but partial steps of the activities are also listed, each with their duration, the generated SQL statements etc. This level is the best setting for the debugging of reports.
- **TRACE:** In the TRACE level all steps of a report execution are shown, down to the loading of single values from Caches or data bases. This level is extremely detailed and can generate many thousand lines for a complex report.

Use the TRACE level only if you really need such a detailed log. Using this levels can generate a lot of data and take accordingly long. Usually the DEBUG level is sufficient enough for analyzing queries.

Unfolding and folding log entries

Expand all



Collapse all



With the buttons "Expand all" and " Collapse all" you can open all entries (and make every single entry visible) or the collapse the complete tree again (then you see only uppermost level of the entries).

The log filter

If you activate the log control panel and set a suitable log level, you can only see the events of the operations which were execute by yourselves within the Workbench - e.g. the execution of reports etc.

However, it is also the possible to view the log outputs of all users who work at the moment with the system. This can be used to debug reports which are executed by other users or in the Web-Frontend.

Filter log session



To deactivate the log filter or to reactivate it later, you must use the button "Disable local session filter" from the tool bar of the tool.

Deleting the log

Clear log



The entries logged up to now can be deleted with the button "Delete log". Deleted entries are lost and cannot be recreated.

Saving the log

Save

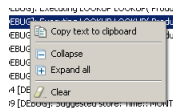


If you want to save the logged entries as a text file on your local file system, you can do this with the button "Save log".

The context menu for log entries

Beside the tool bar there is also a context menu for the single entries of the log. You can be opening this menu with the right mouse button.

The context menu for entries



Beside the functions Collapse, Expand all and Clear described before, the menu offers an other menu item "Copy text to clipboard" with which you can transfer the text of a single entry to the clipboard of your computer.

This function is useful if you like to copy generated SQL statements from the Log and execute them, e.g. to check the result of a single SQL query.

Closing the panel

Close log



To close the panel, you can use the button "Close log" from the tool bar of the tool. Alternatively you also can use the menu item "Tools/Show/Hide log" or the button from the tool bar of the Workbench with which you have opened the tool.

Closing the control panel does not quit the recording of log entries, i.e. a closed log control panel will show all events logged after the closing if you open it again. To switch off the logging you must change the log level on NONE before.

CHAPTER 3:

Using the Query-Editor

Contents of this chapter:

Starting the editor	94
Structure of the editor	97
Editing queries	104

Starting the editor

Queries can be created and edited with the instantOLAP-Workbench. The Workbench (the administration tool of instantOLAP) is a stand-alone desktop application, which is used for remote maintenance and administration of the server. It must be installed on your computer before its use. In order to be able to use the Workbench, you must use a user-account with Power-user or administration rights.

Starting the editor in the Workbench

System-Requirements

In order to create or edit query within the Workbench, it must be installed on your computer and you need an account with the appropriate access-rights of (administration rights) for the instantOLAP server. Read the Workbench documentation for further details.

Creation of new queries

There are three different ways to create a new query within the Workbench: Using the menu, using the toolbar of the Workbench or using the context menu of a folder within the Repository-Explorer.

Creating queries using the menu or the toolbar

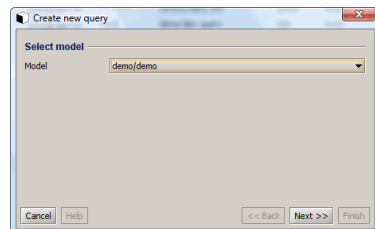
Use the menu-item "Query/New query" in order to create a new query. Then a dialog opens in which you have to specify the model, the folder and the filename and some general properties of the new report.

The new-query button



Alternatively to the menu you can also use the button "New query" from the toolbar of the Workbench.

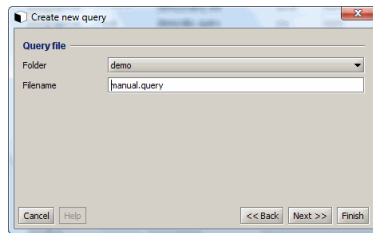
Setting the model



1. Selection of the model: At first you have to determine the model on which your new query will be based. In the drop down field "Model" only the models are shown for which you own sufficient access-rights.

After you have chosen the model press the button "NEXT >>" to proceed with the wizard and to continue with the next page (selection of the folder and filename).

Selection the query-folder and -name

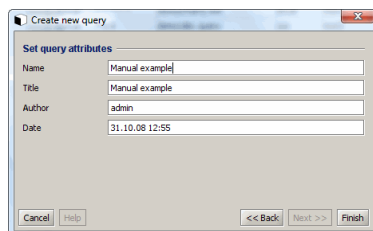


2. Selection of the folder and filename: Contrary to Web-Frontend can you can define a filename for your query and define the folder where it will be stored. An additional page will appear in the wizard after the selection of the model.

Select the folder with "Folder" drop down box and change the filename in the field "Filename" if wanted.

The filename must always end with the extension ".query", otherwise the wizard will not accept the filename and show an error message. You should always give clear names to your queries. Again, confirm the wizard page with the "Next >>" button after you changed the settings (you also may use the "Finish" button to create the report immediately and skip the following page).

Setting the query-attributes



3. General query properties: At last a wizard-page is shown containing the general report properties. You may change the properties at will or keep the default values and change them later in the editor. The following properties can be changed:

- **Name:** The (short) name of the query, which is shown in the navigation of the Web-Frontend.
- **Title:** The (detailed) title of the query, which will be displayed as a title above the report at the time of the report execution.
- **Author:** Name of the author, thus your name.
- **Date:** Time of the query creation.

Confirm your input using the "Finish" button. Now the query will be created and the empty worksheet of the editor will appear.

Creating query using the Repository-Explorer

Instead of using the toolbar you can also use the context menu of the folder in the Repository-Explorer (on the left hand side) in which you would like to create the new query. Use the menu-item "New query". Then the same wizard as above (with a pre-defined folder) appears.

Opening existing queries

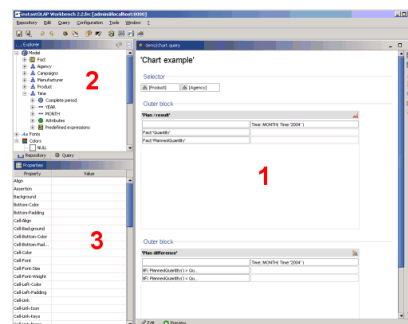
Existing queries can be opened by locating them in the Repository-Explorer (on the left hand side) and using the menu-item "Edit" from their context-menu. Alternatively you can open editor for a query with a double click using the mouse or by dragging the report file onto the desktop-panel of the Workbench.

In order to work on existing queries you must own the appropriate access-rights for it and for the model the query is based on. Otherwise you cannot see the query within the Repository-Explorer or you will receive an error message when trying to edit the report.

Structure of the editor

After creating or editing a new or existing query, the worksheet of the editor appears in the browser or in a separate window within the Workbench.

The editor



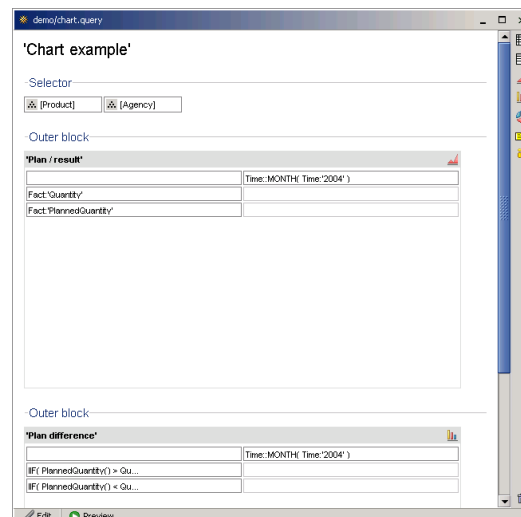
The editor is divided into three different areas:

1. The **Worksheet** representing the query itself,
2. the **Model-Explorer**, with which you can investigate the data model of the query and use its elements for the query with drag&drop,
3. the **Property-Viewer**, which is used to view edit the properties of all elements of a query.

The worksheet

The worksheet is the main work area in the editor, within which you can edit the structure and contents of the query. While editing, it shows a schematic view where all elements of the query are visible and editable but contain no real data.

The Query-Editor worksheet



The worksheet is divided into several, vertically arranged, areas:

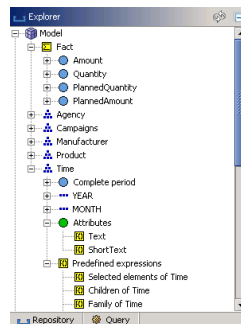
- The title in topmost position,
- the selector area, which is always positioned directly under the title and

- the further sections with the outer blocks, inner blocks, pivot tables and comments. A horizontal line and the text "vertical block" is shown above each outer block.

The Model-Explorer

The Model-Explorer is positioned in the upper left corner of the browser or the Workbench. With this explorer you can investigate the data model used by the query with all its dimensions, dimension levels, facts, attributes, standard formulas etc.

The Model-Explorer



Also, the explorer is the most important source for developing queries using drag&drop. All elements of the model as well as some additional elements (like colors or fonts) can be dragged out of the explorer into the query with the mouse. You can drag this elements onto the worksheet or onto the properties, which are displayed in the Property-Editor.

The Model-Explorer is displayed as in a tree-view (comparable with the File-Explorer from Microsoft Windows). Some elements can contain sub-elements, which can be opened (and closed) by clicking on the plus symbol to the left of the elements.

When working with the Workbench, the Model-Explorer uses the same location as all other explorers. You can switch back and forth between these with the tabulators below the explorer. If you use the editor in the Web-Frontend, only one explorer will be visible and switching is not necessary.

Information for models

When opening the model node (the first entry in the explorer) the following information about the model will be displayed:

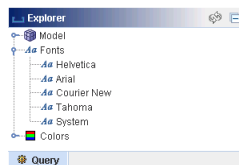
- **Fact:** Under the first node (the sum symbol) all facts of the model are shown. The name of the node is equal to the name of the fact-dimension of the model (an auxiliary dimension in which all facts are stored) and can differ from model to model.
- **All dimensions:** Directly under the fact-node, all dimensions (with the dimension symbol) are shown with their name.
- **The entire hierarchy of a dimension, starting with the rootkey:** For each dimension the root-node of the dimension is shown. By opening this node you can investigate the entire hierarchy of the dimension with all its keys. Keys are marked by the key symbol, a filled out blue circle.
- **All attributes of the dimensions:** Each dimension owns a set of attributes, which you can find under the "attribute" node. All keys of the dimension share these attributes, however not all keys must have values for each attribute. Attributes are marked by the attribute symbol, a filled out green circle.
- **Standard formulas for the dimensions:** The explorer offers some standard formulas for each dimension, which are useful when creating reports. These formulas become visible when opening the node "Predefined of expressions". For formulas, the formula symbol is used.

- **All levels to the dimensions:** Each level of a dimension hierarchy has its own name. All levels with their names are displayed below the dimension node. By opening the level-node (marked by the level symbol) the standard formulas for each level will become visible (see below).
- **Standard formulas for the levels:** As is the case for the dimensions there are standard formulas for levels. These formulas also use the formula symbol.

Fonts

Below the node "Fonts" some standard fonts are offered, which can be used you when creating queries. However, these are not all available fonts - you may use any font being installed on you computer and on the report server.

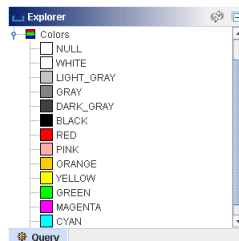
Fonts in the Model-Explorer



Colors

As is the case of fonts, there are also some standard colors under the node "Colors", which can use you in your queries. Also, these are not all available colors you can use but only a small cutout.

Colors in the Model-Explorer



The Property-Editor

The Property-Editor is the table in the left, lower corner of the editor (or of the Workbench). The editor is used for investigating and editing the properties of all elements. Whenever you select an element in the worksheet by clicking on it with the mouse, then all properties of this element are displayed and can be edited.

The table showing the properties

Property	Value
Author	admin
CSS	
Date	30.10.05 14:25
Description	
Filter	
Formats	HTML, EXCEL, PDF, CSV
Icon	
Logo	
Minimum Width	400
Model	demo
Name	Chart example
Print Height	29.7cm
Print Logo	
Print Stylesheet	

Editing properties

In order to change a property, you must click into the value field in the second column, right to the name of the property. After the cursor appears in the table cell, you can edit the value.

Changing properties in the table



Press the "Enter" key or click anywhere outside the cell to confirm the input. Then the system will check the new property value (e.g. for syntax errors in formulas). If an error was detected, it will be displayed in the status line of the browser or Workbench.

Note that you only can see the error message in the browser if the status line of the browser is visible. If you cannot see the status line, you should activate it, e.g. by using the menu-item "View/Status line" in the Internet Explorer.

If no error was detected, the editor will stop the input mode and you will be able to edit other elements afterwards. Otherwise, the input cell will show a red background.

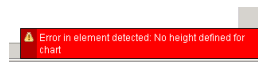
An input containing an error



In the case of an incorrect input you must correct this before you can leave the input field. Alternatively you can use also the "ESC" to reset the property to the previous value and stop editing.

When selecting an element or after changing a property, the system checks all properties of the element and shows an error message if any error was detected. This is useful, because some properties depend on other properties or could have become faulty, e.g. when a dimension was removed and is still referred in the properties.

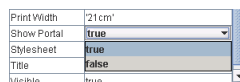
An error message produced by the property checker



Default values

Some properties offer default values from which you one by opening the combo box (the small triangle beside the input field).

Default values for a property



The previous value of the properties is replaced by clicking on a value. After the selection you still can change the property back to its previous value by using of the "ESC" key.

Special editors

For some properties (formulas or colors) the Property-Editor offers special editors, which you can use to edit a value instead of using the simple input-field. If a property offers such an editor, a button containing three points "..." will be visible right to the input field. Press this button in order to edit the property with the editor.

The edit button for properties



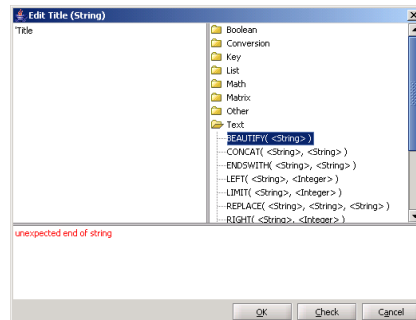
Also, most editors offer a check function, which you can use to examine your input without leaving the editor. Press the button "Check" to examine your input.

The Expression-Editor

The Expression-Editor is probably the most frequently used editor for properties. With this editor you can create and edit formulas, which are used in many properties of the report-elements.

Read the "instantOLAP Reference" for a complete overview of the instantOLAP expression-language and its functions.

The Expression-Editor



The editor is divided into three different areas: The input-area in the upper left area (for editing the text of the formula), the function-table on the right side and the message field at the lower edge.

In the function-table all functions offered by the instantOLAP server (grouped by their topic) are listed. For each function you can see its syntax (i.e. the expected data types of their arguments).

With a double-click on a function, the function will be insert at the current caret-position inside the input-field.

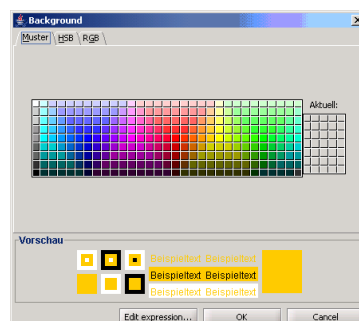
The message field shows possible errors being detected in the current formula.

Use the "Check" button in order to examine your formula on its correctness and to, if any error was found, show the error in the message-field.

The Color-Chooser

The Color-Chooser is used for choosing colors, e.g. for background or font-colors. It offers three different kinds of a selection: You can directly choose a color from a number of pre-defined colors, define a color with its RGB value or define a color using a HSB model (Hue, Saturation and Brightness).

The Color-Chooser








Like other editors, the Color-Chooser has a formula as result, which contains only the selected color as a string constant. If you like to compute a color with a more complex formula, you can change from the Color-Chooser to the Expression-Editor with the button "Edit expression..." at any time.

If you edit a color property which already contains a complex formula, the Expression-Editor will open directly when pressing the edit button of the property.

The toolbar

The Workbench contains a main toolbar with general functions. However, not all of these functions are needed for editing queries. The following buttons are useful when editing queries:







	The "Save" button saves the current query.
	The "Save as" button stores the current query under another filename.
	With the "Undo" button you can undo the last change in your query.
	With the "Redo" button you can redo the last change in your query and revert the last undo operation.
	The "New query" creates a new query.

All functions of the toolbar can also be called from the menu of the Workbench.

The toolbar of the worksheet

Beside the main toolbar for the general functions there is a second toolbar at the right edge of the worksheet, which offers functions for the creation of new elements within the queries.

The following functions are available in this toolbar:

	The "Add pivot-table" button adds you a further pivot table to the query.
	The button "Add line-chart" adds a pivot table to your query, which will be displayed as a line chart.
	The button "Add bar-chart" adds a pivot table to your query, which will be displayed as a bar chart.
	The button "Add pie chart" adds a pivot table to your query, which will be displayed as a pie chart.
	The button "Add comment" adds a comment to the query.
	The "Add list" button adds a list to your query.
	The button "Add newsticker" adds a newsticker to the query.

The preview panel

While editing a query, you can perform a preview of its result at any time without leaving the editor.

Performing a preview

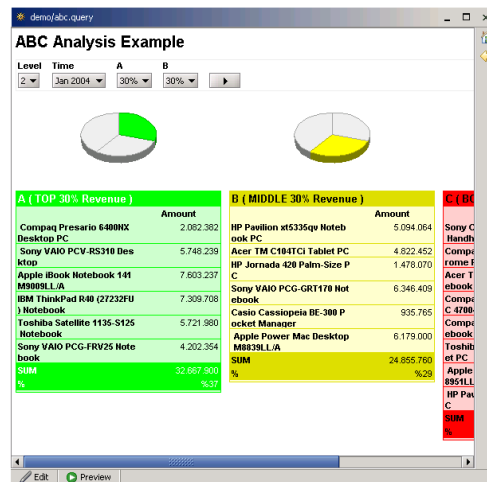
The preview tabulator below the worksheet



In order to switch from the design mode (when the worksheet is visible) into the preview mode, you must use the "Preview" tabulator underneath the worksheet. After switching to the preview, the query will be executed immediately.

After the server successfully processed the query, the result will be displayed in the preview window.

Previewing a query in the Workbench



If the query and the model used by it are error-free, you will see the preview of the query after its execution. Otherwise, an error message will be displayed inside the preview window.

Leaving the preview

The "Edit" tabulator below the worksheet



In order to return to the worksheet, you must use the "Edit" tabulator (left of the "Preview" tabulator).

Editing queries

This section describes how to work with queries and how to create, edit or delete of individual elements.

Storing queries

Queries can be saved at any time using the menu-item "Save" or "Save as" (in the Workbench) or using the "Save" or "Save as" buttons from the toolbar.

Unless you save a query, the current changes are only visible for you and not for other users.

Editing elements

There is generally the same number of possible operations for nearly each element: You can create, move, delete or select most elements and change their property-values.

The only exception is the worksheet, since this can't be created, moved or deleted because the worksheet represents the query itself. But you can edit the properties of the worksheet (and thus the properties of query).

For each element-type its most important properties are described here.

Creating elements

Usually you can use the mouse to create new elements, e.g. with drag&drop out of the Model-Explorer, using a context-menu or button or by copying existing elements.

This section describes how to create new elements for all types of query-elements.

Moving elements

All elements except the worksheet can be positioned (with certain restrictions) freely. This section describes how to move elements for each type.

Deleting elements

Each createable element can also be deleted. The only exception is the worksheet itself.

Selecting elements

All elements of a query (including the worksheet) can be selected in order to change their properties in the Property-Editor. Selected elements are displayed with a blue border.

Changing properties

The properties of the currently selected element can be edited in the Property-Editor in order to change its behavior and appearance.

Nearly all properties of all elements are formulas and are computed at the time of the report execution. Though you also may use simple constants in properties (e.g. if you want to set a

background color to a certain and independent value), you must code this value in the correct syntax for instantOLAP-formulas.

Nearly all properties of all elements are controlled by formulas. Therefore you must use the correct syntax for instantOLAP-formulas - also for constant values. Otherwise the system will show an error message.

E.g. the syntax for a constant blue color would be `"blue"` or `"#0000FF"`, in each case enclosed into simple delimiters. These delimiters mark a constant string in an expression.

Read the "instantOLAP reference" for a detailed description of formulas and their syntax.

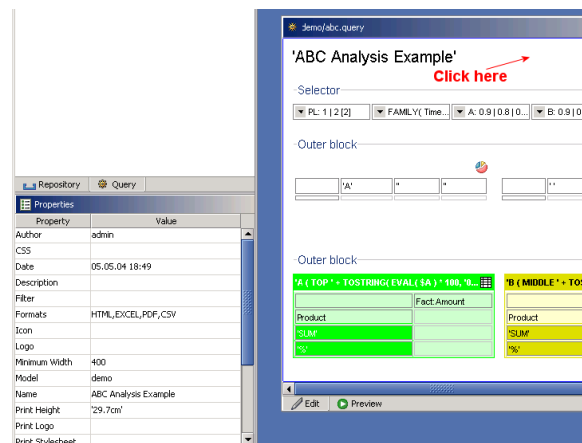
Editing the worksheet

For the worksheet there are only very few possible operations - you can only view and edit the properties of a worksheet.

Viewing and editing the properties

To view the properties of a query must select its worksheet. Click onto the title or any free space of the worksheet to select it.

The properties of a worksheet are displayed after its selection



After you successfully selected the worksheet, it will be displayed with a blue border and its properties will become visible in the Property-Editor.

Setting the query-title

The title of a report can be changed by editing the Title property of the worksheet. You must select the worksheet before editing its properties.

When creating a new query, the Title property will contain the standard title which was defined in the Query-Wizard.

Setting the query name

The name of a query is a constant text which is shown in the index-pages of the Web-Frontend.

In order to change of a query, you must must change the value of the property "Name" of the worksheet.

This Name property expect no formula and you must enter the name without text-delimiters here.

Setting the description

The description of a query is also shown in the index-pages of the Web-Frontend. It is defined by the property "Description".

Like the name of a report, the Description property expects a constant value and no formula. You must enter the description with text-delimiters.

Setting the visibility

Queries can be visible or hidden. Users may still use invisible queries but they won't see them in the navigation of the Web-Frontend. Invisible queries are mostly used a linked detail query and are not directly to be called by the user.

You can change the visibility of a query by changing its property "Visible".

This property expects no formula and therefore only accepts the constant values "true" or "false".

Setting the symbol

The symbol of a query, like its name and the description, is shown in the index-page of the Web-Frontend. By changing the symbol you can replace the standard picture of a query, e.g. with you own symbol a small screen-capture as preview.

To set the new symbol you must change the property Icon of the worksheet.

This property expects a constant value (no formula) which must contain the URL (relative or absolute) of the icon.

It is important that the user has access to the symbol-URL from his workstation. The best way to ensure this is to copy the symbol into the same repository folder as the queries and only use relative URLs in the queries.

Setting the logo

The logo of a query is shown in the upper left corner of the Web-Frontend. You can define a logo both for the index-pages and for individual queries. If no logo was specified, the standard logo "/iolap/logo.gif" will be used.

In order to use a different logo you must edit the property Logo. Since the logo will be only displayed when executing a query, the property can be computed and expects a formulas as its value, contrary to the name, description or symbol of a query.

The URL of a logo must be a relative or absolute URL and always end with the extension ".gif", ".jpg" or ".png".

Beside the logo for the Web-Frontend you can also determine a print logo that will be used when converting a report in to a PDF file. If no print-logo was defined, the system uses the standard logo of the query for the PDF conversion.

Setting the print-format

For the conversion of queries into the PDF format you can define the measures of the pages with the properties Print Width and Print Height.

These properties expect a number followed by a unit, e.g. "cm" or "in". This combination of numbers and text makes it necessarily that you put both in a string, e.g. "'21 cm'".

The most common print-formats can be chosen directly using the the context-menu (see below) of the worksheet without having to change these properties manually.

Setting the stylesheets

instantOLAP uses CSS stylesheets to format the queries when being displayed (in the Web-Frontend). You may adapt these stylesheets and the layout of a query for your needs or use different stylesheets for different queries.

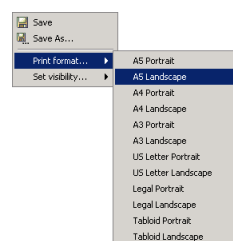
The CSS stylesheet is specified by the property "CSS" of the worksheet. This property expects an URL, which (either relative or absolutely) refers to the CSS file. CSS files usually end with the extension ".css", therefore your URL should likewise end on ".css".

The stylesheet is used at the time of the query execution and is computed with formulas. Therefore this property always expects a formula which has a string as result.

The context menu

The worksheet has its own context menu, which appears when clicking with the right mouse button on any free part of the worksheet (or on its title).

The context menu of a worksheet



Apart from saving the report (with "Save" or "Save as") you can set some properties of the report to pre-defined values here. Those are the print-format and the visibility of the report.

Editing selectors

Viewing and editing properties

In order to edit the properties of a selector you must select it with the mouse before.

After you successfully selected the selector, a blue border will be drawn around the element and its properties will be shown in the Property-Editor.

A selected selector (with blue border)



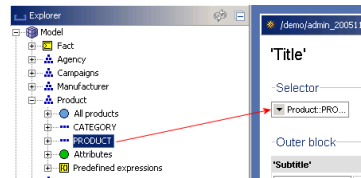
Creating selectors

There are different ways to create a selector:

Creating a selector using drag&drop out of the Model-Explorer

The simplest way to add a new selector to a query is to drag an element out of the Model-Explorer. Navigate to an expression (e.g. a dimension, a dimension-level, an individual key or fact or a formula) inside the explorer and drag this with mouse on a free space within the selector-area.

Create of a selector by dragging an expression out of the Model-Explorer



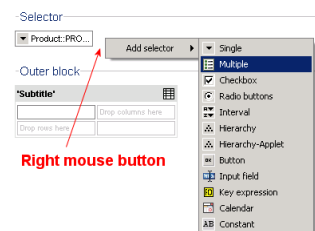
When dropping the expression, a new selector will be added at the end of the selector list (with the expression as its option-expression). The new selector will have the type "Single" or "Hierarchy-Tree" and has no standard selection (you can change both later). You can change the position of the selector within the selector-area afterwards with the mouse (read the section "moving selectors").

The type of the new selector depends on the element you drew out of the Model-Explorer: If you used a dimension, a Hierarchy selector will be created (which allows the user to select a key of a dimension in a hierarchical way). Otherwise a selector of the type "Single" will be created, which allows the selection of an individual key from a list of keys.

Creating selectors using the context menu

You can also create an empty selector manually by using the context menu of the selector-area and set its formulas and options afterwards. Use the right mouse-button to open the menu and choose the menu-item "Add select".

Creating a selector using the context menu



Extending selectors using drag&drop

You may also extend the options of an existing selector with drag&drop. Drag an expression from the Model-Explorer on a selector and drop it there (a blue border will signal that a selector is the target of your drag&drop operation).

Then the Options property of the selector will be extended by the new expression and the selector will offer this new options when executing the query.

Moving a header from a pivot table with drag&drop

A further possibility to create a new selector is to drag a header out of a table into the selector-area. Like dragging an expression out of the Model-Explorer, this will create a new

selector of the type "Single" and without a standard selection. The options of the new selector is equal to the iteration of the header.

Creating a selector from a header



The header used as source disappears after moving it into the selector-area. If you wish to keep the header, press the key "CTRL" while dragging the header.

Copying selectors with drag&drop

In order to copy a selector with drag&drop, you must drag it to the desired position within the selector area, like you would do when moving a selector. While dragging the selector you must keep the key "CTRL" pressed so that the mouse cursor shows the copying symbol (a plus sign).

Copying a selector with drag&drop



Deleting selectors

There are different ways to delete a selector:

Deleting selectors using the keyboard

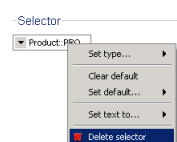
You can use the key "Delete" to remove a selector from the query. First, you must select it with the mouse (a blue border around the selector will show its selection). If you press the key now, the selector will be deleted.

The system will delete the selector without asking you before. If you falsely deleted a selector you can use the "Undo" button from the toolbar to undo the deletion.

Deleting selectors using the context menu

Instead of the keyboard you can also use the context menu of a selector to delete it. Open the context menu using the right mouse button and use the menu item "Delete" to remove the selector.

Deleting a selector using its context menu



Changing the type of a selector

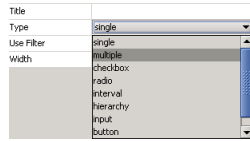
The type of a selector can be changed either by editing its properties or using the context menu:

Changing the type using the properties

In order to change the type you must select the selector first. Click on the selector until a blue border indicates that the selector is currently selected. Now you will see its properties in the Property-Editor.

The property "Type" holds the type of the selector. This property offers a list of all possible selector-types, from which you can select the desired.

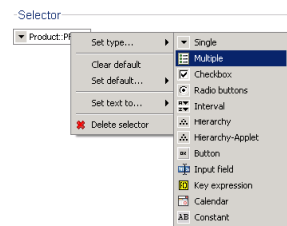
Changing the type of a selector using the property



Changing the type using the context-menu

The faster way to change the type is to use the context menu of a selector. Open the context menu using the right mouse button while holding the mouse over the selector. Now choose the desired type from the sub-menu of the menu-item "Set type...".

Changing the selector type using the context menu



Setting the options

The property Options of a selector determines the options of a selector from which the user can select entries when executing a query (e.g. "every year", months, products etc.). The options are defined with a formula, the result of this formula is converted into the options.

There are two different ways of setting the options: By dragging expressions out of the Model-Explorer or by directly manipulating the property "Options" of the selector.

Extending the options with drag&drop

The Model-Explorer offers different elements of the model and standard formulas, from which you can drag the needed into the selector-area or onto an existing selector to define the selector options.

The options of the selector are then extended by this new expression (the expression is added to the previous expression using the join operator "|").

Changing the options property

You can alternatively edit the property Options manually and change the formula with which the options are computed. Before editing the property you must select the selector-element so that its properties become visible in the Property-Editor. This is the common way to set the options for more complex expressions.

Setting the default selection

The default selection of a selector determines, which element of the options is selected when the user executes the query the first time. If you set no standard selection for a selector (means the property "Default" is empty) the selector will be empty with the first call of the query.

There are two different ways of setting the standard selection: By directly editing the property Default or by using the context-menu of the selector:

Changing the default property

The property Default determines (using a formula) which options of this selector are initially selected. Select the selector (until it is framed with a blue border) and change the property in the Property-Editor.

The count of the default values the formula should return depends of the selector type. Interval selectors need two default values, the first one is used for the first (start) selection and the second one for the second (to) selection.

Make sure that the result of the function is a part of the options.

Using the context menu

Using the context-menu of the selector is a very simple way of setting the default selection for a selector. You can only set the default selection to the first or last option offered by the selector with the menu.

Open the context-menu of the selector by using the right mouse button. Then use the menu-item "Set default..." and choose the first or last options as the new default selection for the selector.

Setting the default selection with the context menu



Changing the default selection using the context menu also changes the property "Default".

Deleting the default selection

You can delete the default selection of at any time, either by deleting the value of the property Default or by using the menu-item "Clear default" from the context menu of the selector.

Setting the title of a selector

The title of a selector is being displayed above the selector in the result. Usually the name of the selector or name of the selectable dimension is used its title.

Alternatively you can assign a different title to a selector. Edit the formula stored in the property Title to change the title. The formula must have a text as result (usually a constant string is used here, e.g. "month" or "product").

Selector with title



If you change the title of a selector, then the formula stored in the "Title" property will be displayed inside the worksheet directly above the selector.

Defining the option-texts

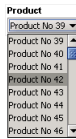
Apart from the title of the selector you can change the texts being displayed for the individual elements (the options) of a selector. A formula is used for the computation of the option-texts.

Usually the IDs of the option-elements are used as text within the selector. By changing the property text you can define a formula which returns different texts for its options.

This formula will be computed for each single option of the selector, whereby the filter for the computation always refers another element of the options (therefore inside the text-formula you can e.g. use the expression "product" to refer the currently selected and displayed product).

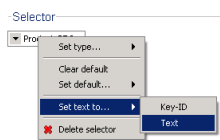
E.g. a formula "Product.ProductNo" would return the product numbers of the products offered as options and the selector would display these product-numbers instead of the product-ids.

A selector with changed option texts



Instead of manually editing this property you can also use the context-menu of the selector to change the text-property. The menu will suggest a list of standard texts.

Setting the options texts with the context menu

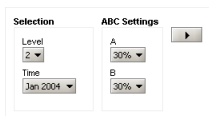


All text attributes of the options are shown here. Which attributes are offered depends on dimensions used in the options-expression, since different dimension own different attributes. Also, the menu could contain useless attributes because they could be technical values (e.g. technical database-IDs of the elements etc.).

Grouping selectors

Multiple selectors can be grouped to a selector-group. This groups arrange all their selectors vertically and there need less space in the selector-area than the selectors would need if they were not grouped.

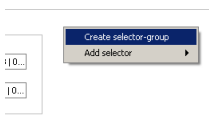
A sample selector-group



Creating a selector-group

To create a selector-group you must open the context-menu of the selector-area. Use the right mouse button above the area to open the menu.

Creating selector-groups using the context-menu of the selector-area

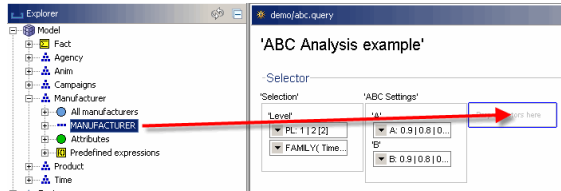


The use the menu item "Create selector-group". Then a new, empty selector-group will be appended to the selector-are.

Adding selectors to selector-groups

Adding new selectors to a selector-group works similarly like adding selectors to the selector-area: You can drag key-expressions from the Model-Explorer or existing selectors or headers into the group. A blue border will signal the moment you can drop the selectors.

Adding selectors to a group using drag& drop

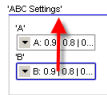


After you dropped the selector it will be added to the group. Depending on the border you dropped the selector, it will be appended at the end or added at the beginning of the group.

Arranging selectors within a group

After you added some selectors to a group you can arrange their order within the group using drag&drop. Select a header and drop it at the upper or lower border of another header inside the group or at the upper or lower border of the group. This will move the selector before or behind the other selector or to the top or bottom of the group.

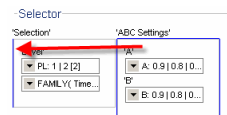
Arranging selectors within their group



Moving selector-groups

You can also move the whole selector-groups and change their position inside the selector area. Moving groups inside the area is like moving single selectors: Select the group and drag it onto the border of another selector or selector-group.

Moving selector-groups using drag&drop



Then the editor will move the group before or behind the other selector or group.

Copying selector-groups with drag&drop

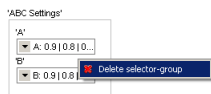
You can also use drag&drop to copy selector-groups. Like when moving a group, drag it to the desired position within the selector-area but keep the CTRL key pressed. Then the editor will create a copy of the group and all its selectors and place it at the target position.

Deleting selector-groups

There are two different ways to delete a selector: Using its context-menu or by using the keyboard.

The context menu of a selector offers a menu-item "Delete selector-group". Open the context menu and use this item to delete a group.

Deleting a selector-group using its context-menu



In order to delete a selector-group with the keyboard you must select it first by clicking onto its title. Then press the delete-key to remove the group from the selector-area.

Changing the appearance of selector-groups

There are several properties for selector-groups which define their appearance in the report:

- The "Width" and "Height" determine the size of the group
- "Title" defines the title of the group. If no title is specified, it will remain empty
- "Background" defines the background-color of the group
- "Border" sets the border-color of the group

A selector-group with title, background- and border-color



The context-menu for selectors

Selectors have, as mentioned before, their own context-menu, with which different operations on the selector can be performed:

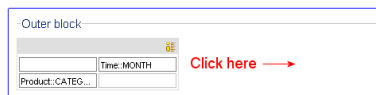
- Changing the type of a selector
- Setting and resetting the standard selection of a selector
- Setting the option-texts to a certain attribute
- Deleting selectors

Editing outer blocks

Viewing and editing the properties

In order to view the properties of an outer block you must select the block with the mouse first. To select an outer block you must click onto its title or onto the free space under the dividing line of the outer block.

Select an outer block by clicking into its free space or title



After you successfully selected the block, a blue border will be drawn around the block and its properties will be displayed in the Property-Editor.

Creating outer blocks

There are different ways to create an outer block: Using the toolbar of the Query-Editor or by copying existing blocks within the query.

Creating blocks using the toolbar

Outer blocks cannot be created directly using the toolbar since no button is available for this operation. However, you can create a table or chart by using the toolbar which will also create an inner and outer block for this new element.

The "New table" button



E.g. use the "New table" button to create a new pivot table. Then an outer and inner block would be created together with the pivot-table.

Copying an existing blocks with drag&drop

Existing blocks can be copied with drag&drop. Select the block you want to copy and drag this, while pressing the "CTRL" key, to the desired position. Whenever the mouse is located on a valid target-position, a blue line will become visible in the worksheet and indicate this. Now you can drop the block here and create a copy of it (the copy will also contain copies of all nested inner blocks and pivot tables, too).

Moving outer blocks

Existing block can also be moved with drag&drop. Select the block and drag it, using the mouse, to the desired position. Again, a blue line in the worksheet indicates valid target-positions.

Deleting outer blocks

Outer blocks can be deleted using the keyboard or with their context menu:

Deleting blocks using the keyboard

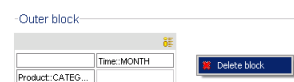
You can use the "Delete" key to remove an existing outer block. Before that, you must select the block with the mouse until a blue border is drawn around it. The press the key and the block will be deleted.

The system will delete the block without asking before. If you falsely deleted a block you can use the "Undo" button from the toolbar to undo the deletion.

Deleting blocks using their context menu

Alternatively you can use the context menu of an outer block to delete it. Select the block and open its context menu using the right mouse button. Then use the menu item "Delete block".

Deleting a block with its context menu



Setting the iteration

For blocks (both for outer and inner block) you can define an iteration. A block with an iteration will be repeated for each key which is part of the result of the formula stored in the Iteration property.

The formula must return a number of dimension-elements (key) and the block will be repeated for each key which is part of the result of the formula.

An outer block iterated over the manufacturer dimension

Jan 2004			
	Quantity	Planned	Amount
Compaq Presario 6400NX Desktop PC	2.233	2.254	2.082.382
Feb 2004			
	Quantity	Planned	Amount
Compaq Presario 6400NX Desktop PC	2.023	2.063	2.013.312
Mrz 2004			
	Quantity	Planned	Amount
Compaq Presario 6400NX Desktop PC	2.104	2.153	2.019.925
Apr 2004			
	Quantity	Planned	Amount
Compaq Presario 6400NX Desktop PC	1.976	1.948	1.894.085

Setting the iteration using the property

The most direct way to set the iteration is to edit the formula stored in the property "Iteration". This property expects a formula with dimension-keys as its result.

Setting the iteration using drag&drop

You can also edit the iteration by dragging an element (e.g. a dimension level or any other expression) from the Model-Explorer into the block. Then the iteration of the block will be set to this expression or it will be extended (if the iteration is already set) with the new expression.

The context menu for outer blocks

All blocks own, as mentioned before, a context menu. The only operation offered by this menu is the deletion of an existing block.

Editing inner blocks

Viewing and editing properties

In order view the properties of an inner block you must select the block with the mouse first. To select an inner block, you must click on its title.

Select an inner block by clicking onto its title



After you successfully selected the block, a blue border will be drawn around it and its properties will be shown in the Property-Editor.

Properties for charts

If you display a block as a chart (when using a chart-format as the display format for a block), a set of additional properties will become visible in the Property-Editor. This properties define the layout and the presentation-form of the chart.

Creating inner blocks

There are different methods to create an inner block: You can use the toolbar of the Query-Editor or you can copy existing blocks within the query.

Creating blocks using the toolbar

Inner blocks cannot be create directly with the toolbar, since no button for creating inner blocks exists. However, you may create a table (or chart) using the toolbar which also will create a new outer and inner block.

The "Add table" button



E.g. use "Add table" button to create a new pivot table. Then an outer and inner block would be created together with the pivot-table.

Copying blocks using drag&drop

Existing inner blocks can copied with drag&drop. Select the block and then drag it, with pressed "CTRL" key, to the desired position. Whenever the mouse is located on a valid target-position

for the block, a blue line will become visible in the worksheet. Now you can drop the block here and create a copy of it (the copy also will contain copies of all nested pivot-tables, too).

Inner blocks can be copied both into existing outer blocks and directly onto the worksheet of the query. A new outer block will be automatically created when dragging an inner block onto the worksheet.

Moving inner blocks

Existing block can also be moved with drag&drop. Select the block and drag it, using the mouse, to the desired position. Again, a blue line in the worksheet indicates valid target-positions.

Deleting of inner blocks

Inner blocks can be deleted using the keyboard or by using their context menu:

Deleting blocks using the keyboard

You can use the "Delete" key to remove an existing inner block. Before that, you must select the block with the mouse, until a blue border is drawn around it. Then press the key and the block will be deleted.

The system will delete the block without asking before. If you falsely deleted a block you can use the "Undo" button from the toolbar to undo the deletion.

Deleting blocks using their context menu

Alternatively you can use the context menu of an inner in order block to delete it. Select the block and open its context menu using the right mouse button. Then select the menu item "Delete block" to delete the block.

Delete a block by its context menu



Setting the iteration

For blocks (both for outer and inner block) you can define an iteration. A block with an iteration will be repeated for each key which is part of the result of the formula stored in the Iteration property.

The formula must return a number of dimension-elements (key) and the block will be repeated for each key which is part of the result of the formula.

An inner block which repeated over the months

Acer			Apple		
	Quantity	Amount	Quantity	Amount	
Acer TravelMate 2330V Notebook	51.436	136.610.138	Apple eMac Desktop 17 M8951LL/A	51.028	118.2
Acer TM C104TC1 Tablet PC	59.958	146.150.492	Apple Power Mac Desktop M8638LL/A	51.748	163.7
			Apple iBook Notebook 141 M9003LL/A	59.802	187.3

Setting the iteration using the property

The direct way to set the iteration is to edit the formula stored in the property "Iteration". This property expects a formula with dimension-keys as result.

Setting the iteration using drag&drop

You can also edit the iteration by dragging an element (e.g. a dimension level or any other expression) from the Model-Explorer into the block. Then the iteration of the block will be set to this expression or it will be extended (if the iteration is already set) with the new expression.

Setting the block title and -font

Inner block can have sub-titles which are displayed above their contents. To define such a sub-title you must set the property "Title" to a appropriate formula.

You may also set the font and size for the title. Edit the properties "Font" and "Font Size" to change them.

Setting the frame-color

A frame is drawn around the block, the color of this frame is always equal to the background color of the block title. If you want to change the frame color, you can define the color using the property Border (which expects a formula returning a color-name or the RGB value of the color).

An inner block with black frame

Complete period	Quantity	Amount
Acer	111.394	284.760.630
Apple	153.578	469.249.188
Casio	51.438	22.425.776
Compaq	211.480	330.706.080
HP	158.306	224.483.638
IBM	51.638	171.743.184
Sony	203.616	432.825.626
Toshiba	109.136	305.099.026

If you want to make the frame invisible, leaving the property "Border" empty would not help - the frame is painted with the standard color for frames then. Instead you must set the frame color equal to the background color of the query (usually "white").

Setting the block format

The block format of an inner block determines the representational form of the pivot tables contained in it. Inner blocks may display their contents as tables, charts or newsticker.

Setting the block format by editing the format property

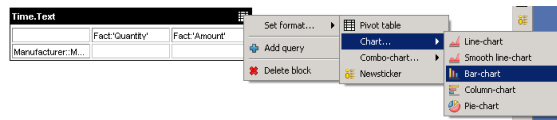
The property format of an inner block determines the representational form of its pivot tables. This property does not expect a formula, only a constant string with one of the following values:

- **table** displays the pivot tables as a simple table. This is the standard display format for blocks.
- **easychart** displays the pivot tables as charts. With the second property "Chartformat" you can define the type (line, bar, pie or spline) of the chart.
- **newsticker** represents pivot tables as a newsticker.

Setting the block format using the context menu

With the context menu of an inner block you can determine its display format. All charts types mentioned above are offered as menu-items. For charts, all charts type are available as menu-items under the items and "Combo Chart".

Setting the block format with the context menu



When setting a chart format using the context-menu, some chart properties will be set to predefined values which can save you some work. E.g. the block size, the legend settings and other properties will be set.

Always set the format for inner blocks with their context menu, especially if you want to use charts. Then, dependent on the chart type, many properties of a block and chart are initialized with default values.

The current format of an inner block is always shown as a small symbol in the upper right corner of the block.

The format of a block (here a bar chart) is displayed as symbol in the upper right corner



Setting the block size

The properties Width and Height define the size of an inner block in pixels.

For blocks, whose contents is formatted as a simple table, you do not have to set these properties, since for these the size can be computed automatically by the system (although setting their size is still possible and will change the appearance of the block).

However, if the block is displayed as a chart, this properties must be set, since they also determine the size of the chart. If you have set the chart format of a block using its context menu, then size will be already set to default values.

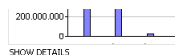
Always set the size for inner blocks, whose contents is displayed as a chart. For this use the properties Width and Height.

Providing a link for a block

For inner blocks you can define links in order to refer to other report or URLs. The URL of a link must be set with the property Link. When the formula in this property returns a value, the link will be displayed underneath the block in the result.

Optionally you also add a name and an icon to the link by editing the properties "Link Name" and "Link Icon". The icon will be displayed next to the link text. The name will be (if defined) displayed instead of the link URL.

A block with an additional link



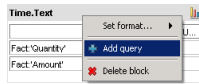
Adding further pivot tables to an inner block

Usually inner blocks contain only one pivot table.

However, for overlay charts (combined charts which consist e.g. of a bar and a line chart) it is necessary to define more than one pivot table in a single block. Each pivot table is then converted into a single layer of the overlay chart.

Use the context menu of the inner block to add a new pivot table to it. Open the menu using the right mouse button and then use the menu item "Add query" to add a new pivot-table to the block.

Creating a new pivot table using the context menu of the block



The context menu for internal blocks

Inner block own a context menu offering the following operations:

- Changing the display-format of the block
- Adding further pivot tables to the block
- Deletion of the block

Editing pivot-tables

Viewing and editing the properties of a table

In order to view and edit the properties of a table you must select it with the mouse before.

Since a table is not visible (only the corner and the headers of a table are visible), you must click on the upper, left corner of the pivot table to select it.

Click on the upper left corner of a pivot table to select it

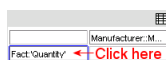


After you successfully selected a table, a blue border will be painted around the corner of the table and the table properties will be displayed in the Property-Editor.

Viewing and editing properties of an individual header

Headers are placed within a pivot table and can also be selected. Click directly on a header in order to select it and view its properties in the Property-Editor.

Click directly on a header to select it



Creating new pivot tables

There are several ways to create a new pivot table:

Creating pivot tables using the toolbar

The common way to create a new pivot table is to use the button "New table" from the toolbar of the Query-Editor.

The "Add table" button

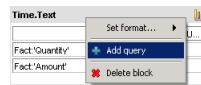


Use the button "Add table" to create a new pivot table. When creating a table, an additional outer and inner block are created and contain the new pivot-table.

Creating pivot tables using the context menu of the inner block

If you want to create an additional pivot table in an existing block you can use the context menu of the inner block. Select the block first and open its context menu using the right mouse button. Then you can create a new pivot table with the menu item "Add query".

Adding a new pivot table to a block using the context menu of the block



Usually you will not create pivot tables without a block. Only for creating combo chart there is a need for additional pivot tables in an existing block. Then use the context menu to create the (see below).

Creating and moving pivot tables with drag&drop

Pivot tables cannot be moved or copied using drag&drop. But you may move or copy the inner block containing the pivot table in order to move or copy the table with it.

Read the sections Editing outer blocks and Editing inner blocks to learn how to copy or move blocks with drag&drop.

Deleting pivot tables

Deleting pivot tables together with the containing blocks

Usually you wont delete a pivot table directly but together with the inner our outer block containing the table. To delete a block you must select it first and then press the "Delete" key or use the context menu of the block.

The system will delete the table without asking before. If you falsely deleted a table you can use the "Undo" button from the toolbar to undo the deletion.

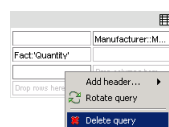
Deleting pivot tables using the context menu

You can also delete individual tables from an inner block until only the empty block remains.

For this select the pivot table you want to delete by selecting its upper left corner and then open its context menu using the right mouse button.

Use the menu item "Delete query" in order to remove a pivot table from its containing block.

Use the context menu to remove a table from its block



Creating headers inside a pivot table

Adding headers to a pivot tables is one of the main tasks when using the Query-Editor. The headers of a table determine the structure and the contents of the pivot tables and therefore of the entire query.

There are different methods to add new headers to a table: By dragging elements out of the Model-Explorer, using the context menu of the table corner or other headers or by dragging existing headers.

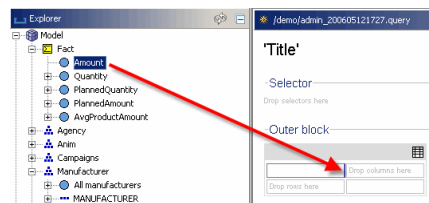
Creating headers using the Model-Explorer

The common way to add a new header to a pivot table is to drag an element out of the Model-Explorer. In the Model-Explorer you can find the mostly used elements of a model, including dimensions, dimension levels, dimension elements and facts, with which a query can be built up.

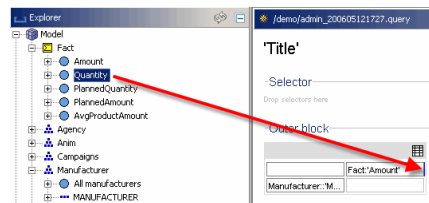
Drag the element you want to use out of the explorer and drop it at the position inside the pivot table where you want to create a new header. When placing elements you have to pay attention to place the elements at one of the edges of the table corner or of an existing header. Depending on the edge, the new header will be added and nested to table table in different ways:

- If you drop the new header at the right border of the table corner or at the left or right edge of an existing header in the x axis, then the new header will be added on the x-axis next to the existing header.

Drop new elements at the edge of an empty table

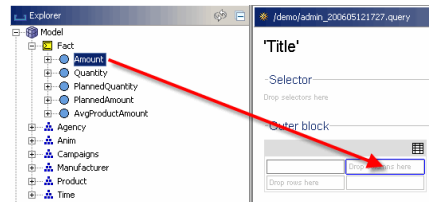


Extend the x axis at the right or left border of an existing header



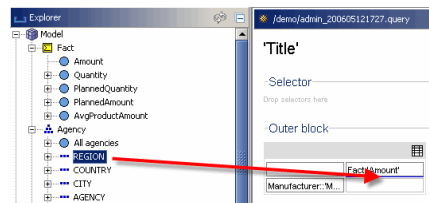
- On empty tables, you can also drop a header in the center of the placeholder named "Drop columns here".

Drop headers on empty tables on the placeholder



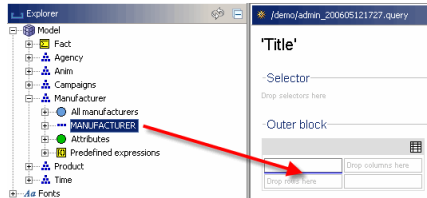
- If you drop the new header at the upper or lower border of an existing header in the x-axis, the new header will be nested under the existing one or the existing header will be nested under the new one (which then takes over the position of the existing header).

New headers can be nested under existing ones

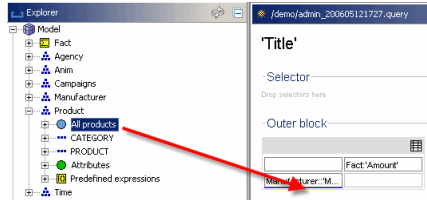


- If you drop the new header at the lower border of the table corner or at the upper or lower edge of an existing header in the y axis, then the new header will be added on the y-axis next to the existing header.

Drop new elements at the edge of an empty table

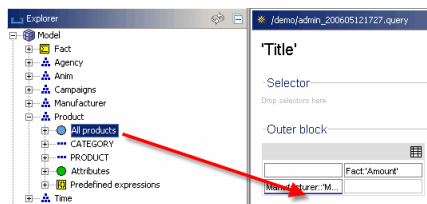


Extend the y axis at the top or bottom border of an existing header



- On empty tables, you can also drop a header in the center of the placeholder named "Drop columns here"

Drop headers on empty tables on the placeholder



If you drop the new header at the left or right border of an existing header in the y-axis, the new header will be nested under the existing one or the existing header will be nested under the new one (which then takes over the position of the existing header).

New headers can be nested under existing ones



Whether a new header may be dropped at the desired position will be always indicated by a blue line at a valid position.

Creating headers using drag&drop within the table

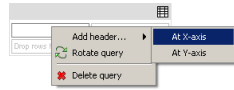
Not only the Model-Explorer can be used as source for drag&drop when creating a header, but also existing headers and selectors. To copy an existing header, you must press the key "CTRL" while dragging it to the desired position. Pressing the key changes drag&drop from the "move" the "copy" mode, which is indicated by a little plus sign in the mouse-cursor.

When using a selector as source, its options expression will be used as the iteration for the new header. When using another header as source, all properties of the existing header will be copied with it. Nested header will not be copied with a header, therefore you only can copy single headers.

Creating headers using the context menu of the corner

You can add empty header to a pivot table by using the context menu of its corner. Open the menu by using the right mouse button and use the menu item "Add header/At X-Axis" or "Add header/At Y-Axis" in order to create a new header on the x or y axis of the table.

The context menu of the corner can be used to create headers

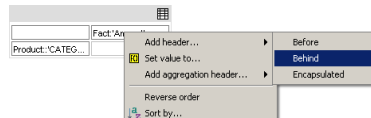


If the appropriate axis already contains headers, the new header will be inserted before all other headers and becomes the first header in the axis.

Creating headers using the context menu of other headers

You can also create new headers using the context menu of existing headers. Open the menu and use one of the menu items "Add header/Before", ".../Behind" or ".../Encapsulated".

Use the context menu of existing headers to create new ones



The new empty header will be created and inserted, depending the used menu item, before, behind or nested to the existing header.

Also you can create "aggregation headers" with the context menu. Aggregated headers will contain a formula which will compute e.g. the sum or average for a table row or column.

Moving headers

You can change the position of a header inside a table with drag&drop. For this, drag a header with the mouse to its new position.

Like when copying headers with drag&drop, the exact target position at which you drop the header will define where and how the new header is located (see above).

After moving a header it disappears from its old position. Its nested headers will then take over the old position of the header.

Deleting headers

Headers can be deleted in two different ways, using the keyboard or using their context menu. The deletion of an header only deletes the header itself, all nested headers remain and will take over the position of the deleted header.

Deleting headers using the keyboard

First select a header with the mouse. Then delete the header with the "Delete" key.

Deleting headers using their context menu

Header own a context menu which you can open using the right mouse button.

Use the menu item "Delete" from its context menu in order to delete a header.

Setting the contents and layout of the table corner

Setting the corner-text and -font

The corner of a pivot table can contain a text which is displayed in the upper left corner of the table. This text can be defined with the property Corner Text of the table.

Displaying a text in the corner of a table

Sales		
All amounts in EURO	Quantity	Amount
Desktop	258.260	524.724.708
Notebook	409.184	1.098.361.324
Pocket PC	203.520	122.257.350
Tablet PC	179.222	495.949.726

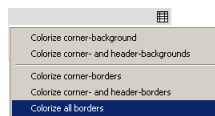
You can also set the color, font and font-size of the corner by changing the properties Corner Foreground, Corner Font, Corner Font Weight and Corner Font Size.

Setting the border- and background-color of the corner

Like for headers you can change the background color and the border color for each single edge of the corner. The background is defined by the property Corner Background, the border color by the four properties "Corner Top Color", "Corner Bottom Color", "Corner Left Color" and "Corner Right Color".

A faster than editing this properties is to drag a color out of the Model-Explorer and to drop it on the corner. You can either change the color of a border directly by targeting them exactly with mouse or you can drop the color in the center of the corner. Then a popup menu will appear, offering different options for the coloration of the corner.

Colorize the corner's borders by dropping a color on it



Defining the layout for headers

For headers there is a large number of layout possibilities. Each header offers properties which define the layout of the header itself and properties for the cells generated by the header at runtime. This means, the layout of the cells will be defined by the header next to it or above it and not by the cells (which do not exist at design time).

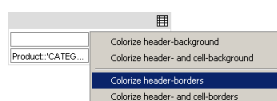
Setting the border colors for headers and cells

The border colors for headers are defined with the properties "Left Color", "Right Color", "Top Color" and "Bottom Color". The border colors for the cells generated by an header are defined with the properties "Cell Left Color", "Cell Right Color", "Cell Top Color" and "Cell Bottom Color".

Like with the corner, you can also use drag&drop to add colors to the borders of headers and avoid to edit their properties directly. Drag a color out of the Model-Explorer and place it exactly on the border of the header you want to set a color for. This will set the color of the border for the header and for the cells generated by this header.

When dropping a color in the center of a header, a popup menu will open and offer some operations for colorizing headers. E.g. use the menu-item "Colorize header-borders" to set the color for all four borders of a header at once.

The popup menu used to colorize headers



If you don't want to set the border color for each single header you can also drop a color on the corner of the table (as described above). Then a menu will appear with which you also can colorize all headers of a table at the same time.

Setting the background colors

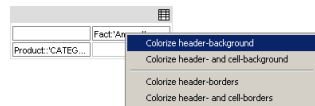
The background color for a headers is defined by its property "Background", the background color of the cells generated by the header is defined by its property "Cell Background".

Using colors for backgrounds and borders

	Amount
Desktop	524.724.708
Notebook	1.098.361.324
Pocket PC	122.257.350
Tablet PC	495.949.726

You can use drag&drop to colorize header backgrounds. Drag a color out of the Model-Explorer and drop in the center of a header. Now a popup menu appears which offers several operations for the colorization of header backgrounds.

The popup menu used to colorize headers



If you don't want to set the background color for each single header you can also drop a color on the corner of a table (as described above). Then a menu will appear with which you also can colorize all backgrounds at the same time.

Setting the font, text size and text color

With the properties Font and Cell Font you can determine the used font for headers and their cells, the font weight is determined by the properties "Font Weight" and "Cell Font Weight". Also, the font and font-weight for the corner can be changed with the properties "Corner Font" and "Corner Font Weight". To change the size of the fonts you must define them (in pixels) with the properties "Font Size" and "Cell Font Size". The color of the texts is defined by the properties Color (both for headers and the corner) and Cell Color.

A changed font, font-size and font-color in a table

	Amount
Desktop	524.724.708
Notebook	1.098.361.324
Pocket PC	122.257.350
Tablet PC	495.949.726

You may also use the pre-defined fonts in the Model-Explorer with drag&drop to set the fonts of the headers and their cells. Simply drag the fonts out of the explorer on the headers.

Setting the text alignment

You can change the text alignment (both horizontally and vertically) for the corner, headers and for the cells generated by a header. Use the properties "Corner-Align", "Corner-VAlign", "Align", "Cell Align", "VAlign" and "Cell VAlign" to set the alignments.

If you do not define the alignment, then this default values are used for the alignment of corners, headers and cells:

- The corner is always aligned top / left
- Headers on the x axis use the alignment 'center'
- Headers on the y axis use the alignment 'left'
- Cells are always are right aligned

The fastest method to change the alignment of a header is to use its context menu. Open the context menu and use one of the four menu items "Align text...", "Align cell-text...", "Vertical align text..." and "Vertical align cell-text..." to set the alignment.

Rotating headers

In order to save place in the result of a query you can rotate header by 90 degrees. This particularly can compact headers on the x axis of a pivot table.

Rotated headers in a pivot table

	Graz	Salzburg	Brno	Bern	Geneve	Berlin	Frankfurt a.M.	München
Desktop	7.550.215	8.864.104	9.830.270	8.046.594	9.917.670	22.393.728	26.261.666	25.977.144
Notebook	16.708.498	19.923.380	20.866.400	17.620.276	22.986.674	47.643.268	56.454.426	58.678.144
Pocket PC	1.801.178	2.150.626	2.364.698	1.906.974	2.417.526	4.988.886	6.425.348	6.463.144
Tablet PC	7.265.586	8.749.280	9.070.726	8.111.628	9.883.846	21.528.576	25.987.750	27.144.144

There are two different ways to rotate headers: By setting the property "Rotate" of a header to "true" or by using its context menu. There you can use the menu item "Text orientation" to switch the rotation of a header on or off.

Using number formats

The number format of a header defines the representation of numbers, percentages or dates in their cells. If you set no format, then the following standard formats for the cells are used:

- If a fact is used as an iteration for a header, then the standard format of the fact will be used in its cells (presumed the fact has a standard format).
- Otherwise the standard format of the displayed data type is used.

You may also set an explicit format for a header to format a complete table row or column. You will particularly need this to format cells being computed with formulas (then no fact is used as iteration and no standard format will be available).

The format is determined with Format property of the header. This property expects a valid number- or date-format of instantOLAP. You find a description of formats with examples in the "instantOLAP Reference".

The alternative way to specify a format is to use the context menu of the header. Here you find pre-defined formats under the menu-item "Set format to...".

Visibility of headers and cells

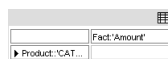
Headers and cells can be made invisible in order to let so individual cells or whole rows or columns disappear. Use the Visible property or the context menu of a header (the menu item "Visibility...") to change their visibility.

Activating drilldown

You can use the property Drilldown to allow the the hierarchical navigation in a dimension.

This property must be edited for each individual header, because headers do not offer drilldown by default. To enable the drilldown, you must set the property to "true" or use the context menu of a header (the menu item "Set drilldown...").

A header with activated drilldown (design view)



There are two different layouts for drilldown, a vertical and a encapsulated layout. With the vertical layout the unfolded headers appear under the original header, with the encapsulated their appear nested under the original header. To activate the encapsulated layout (the vertical layout is the default-setting) you must set the property "Drilldown-Encapsulated" to "true". When

activated drilldown using the context menu of a header, you can choose between these two types.

Vertical drilldown

	Amount
Desktop	524.724.708
▼ Notebook	1.098.361.524
Acer TravelMate 233CV Notebook	136.610.138
Apple iBook Notebook 141 M9089LL/A	187.316.766
Compaq Presario 1215 Notebook	83.587.564
HP Pavilion x5335qv Notebook PC	126.021.524
IBM ThinkPad R40 (27232F1) Notebook	171.743.184
Sony VAIO PCG-FRX25 Notebook	102.129.680
Sony VAIO PCG-GR1178 Notebook	163.092.202
Toshiba Satellite 1135-S125 Notebook	127.860.266
▶ Pocket PC	122.257.350
▶ Tablet PC	495.949.726

Encapsulated drilldown

	Amount
Desktop	524.724.708
▼ Notebook	1.098.361.524
Acer TravelMate 233CV Notebook	136.610.138
Apple iBook Notebook 141 M9089LL/A	187.316.766
Compaq Presario 1215 Notebook	83.587.564
HP Pavilion x5335qv Notebook PC	126.021.524
IBM ThinkPad R40 (27232F1) Notebook	171.743.184
Sony VAIO PCG-FRX25 Notebook	102.129.680
Sony VAIO PCG-GR1178 Notebook	163.092.202
Toshiba Satellite 1135-S125 Notebook	127.860.266
▶ Pocket PC	122.257.350
▶ Tablet PC	495.949.726

Apart from the layout you can also define the behavior of the drilldown with other properties: The property Drilldown-Iteration defines which dimension elements are shown when unfolding a header (usually this will be the children of the element being displayed in the header). With the property Drilldown-Prefetch you can determine if the drilldown-symbol of a header is displayed always, independent of the number of keys being displayed when unfolding a header (in some case this can speed up the report).

*When using the vertical layout for drilldown you can define a padding for the unfolded headers based on their nesting level. Use the property "Left Padding" of the header to define such a padding and e.g. set the property to the formula "DRILLLEVEL() * 10". When activating a vertical drilldown using the context menu of a header, this formula will be used automatically for this property.*

Sorting pivot tables

Often the headers of a table should be sorted by a criterion (usually a fact) - more exactly the headers generated by a header iteration should be sorted. There are different possibilities to define the assortment of a header: You can use their properties "Sort" and "Sort Descending", the context menu of a header or use the SORT function (for top 10 queries) in their Iteration property.

Using the properties Sort and Sort Descending

With the property "Sort" you can define a sort criterion for a header which is used when sorting the headers generated by a header-iteration. Make sure this formula only returns one value for each header, otherwise the system will report the error message "Ambiguous sort value" (if more than one value would exist for any element, sorting would be impossible).

Usually the headers are sorted ascending to their sorting value. If you wish to sort the header descending, you must set its property "Sort Descending" to "true".

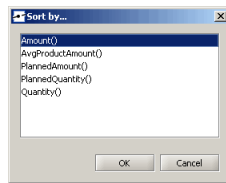
Headers sorted by the fact "Amount"

	Amount
Apple iBook Notebook 141 M9089LL/A	187.316.766
Toshiba Portege 3385 Tablet PC	177.238.760
IBM ThinkPad R40 (27232F1) Notebook	171.743.184
Compaq TC1000T Tablet PC 470045-149	170.560.474
Apple Power Mac Desktop M8839LL/A	163.731.764
Sony VAIO PCG-GR1178 Notebook	163.092.202
Acer TM C1041Ci Tablet PC	148.150.492
Acer TravelMate 233CV Notebook	136.610.138
Sony VAIO PCV-R5310 Desktop	132.972.196

Using the context menu

Usually headers are sorted by a fact. You can use the context menu of a header to select a fact with the menu item "Sort...".

The sort-dialog opened by the context-menu of the header



With the menu item "Sort order..." you can choose between the ascending and descending sort order.

This menu items manipulate the properties "Sort" and "Sort Descending" mentioned above.

Top-10 queries

For top 10 queries it's not enough to specify the assortment of a header since you only want to display a part of its iteration result (e.g. the 10 products with the most turnaround). Therefore you must manipulate its iteration and use the SORT function, which allows to specify a sort criterion and a limit for the size of its result.

The simplest way to create such a limited search is to use the context menu of the header. Select a fact from the menu item "Top 10...". Then the property "Iteration" will be changed and use the SORT function. You can change the limit for the SORT function later (the default limit used for the SORT function is 10) to generate less or more than 10 iterations.

Suppressing blank rows and columns

When creating queries it is often necessary to remove blank rows or columns from the result since only a few cells may contain values. Particularly nesting headers can produce sparse filled tables.

There are different ways to eliminate blank rows and columns from reports: Using the properties "Suppress" Rows and "Suppress Columns", using LOOKUPS and using subcubes:

Suppression using properties Suppress Rows and Suppress Columns

The simplest way to remove blank rows or columns from a pivot table is set the properties "Suppress Rows" or "Suppress Columns" of the table to "true". Then, the empty rows or columns will be deleted from the result.

*Since version 2.5 of instantOLAP, empty rows will be deleted **while** the processing of the query, so there the only limitation for rows is now the number of **displayed** rows.*

Suppression using the LOOKUP function

You should use the LOOKUP function if you want to remove a large number of blank rows from a pivot-table. This function computes the keys of a dimension with present data for a certain day, customer etc. (considering the current filter of the pivot table).

When using the LOOKUP function in the iteration of a header, its iteration will be limited to those dimension elements with data before the pivot table is generated in the memory of the server and the table remains slim.

The simplest way to create an iteration using a LOOKUP function is to use the context menu of a header. Select the fact you want to perform a lookup for from the menu item "Lookup...". Then the iteration of the header will be manipulated and use the function.

Suppression using subcubes

The more complex but also more powerful way to eliminate blank rows and columns from a pivot table is to create a subcube with the property Subcube. A subcube limits all dimensions for an individual pivot table to a subset of their elements and all headers of table will be limited to this elements.

In the Subcube property you can also use the LOOKUP function mentioned above to reduce the dimensions the their part providing data. This will concern all headers of a pivot table automatically. Read the section Filters and subcubes for further information.

The context menu for pivot tables

The context menu for pivot tables offers the following operations:

- Creation of new headers
- Rotating pivot tables
- Deletion of pivot tables

You can open the context menu of pivot tables using the right mouse button on the upper, left corner of the pivot table.

The context menu for headers

The context menu for headers offers the following operations:

- Creation of new headers and aggregating headers
- Sorting and top-10 reports
- Alignment for header texts and cell contents
- Formatting of cell contents
- Deletion of headers

Editing comments

Viewing and editing the properties

Select a comment with the mouse to view and edit its properties in the Property-Editor. Since a comment is not visible (only its text) you must click onto the free space under the separator line of the comment.

Click on the title or the free space next to a comment to select it



After you successfully selected the comment, a blue border will be drawn around it and its properties will be displayed in the Property-Editor.

Creating comments

The only way to create a new comment is to use the toolbar of the Query-Editor.

The "Add comment" button



Use the button "Add comment" from the toolbar in order to append a new comment to your query. The new comment will be appended at the end of the worksheet and can be moved to the desired position afterwards.

Moving and copying comments

Comments can be moved inside a query with drag&drop. First select the comment (by clicking on the free space under its separator line) and move it with the mouse. A blue line indicate valid target positions for the comment.

Copying comments with drag&drop works exactly the same, but you must press the "CTRL" key while moving the comment. The mouse pointer will show a little plus symbol then. When dropping the comment at the target position, a copy of it will be created and the comment does not disappear from its original position.

Deleting comments

There are two different ways to delete a comment, using the keyboard or using the context menu of a comment:

Deleting comments using the the keyboard

In order to delete a comment you can use the "Delete" key. First you must select the comment with the mouse. Then press the "Delete" key to delete the comment.

The system will delete the comment without asking before. If you falsely deleted a comment you can use the "Undo" button from the toolbar to undo the deletion.

Deleting comments using their context menu

Alternatively you can use the context menu of a comment to delete it. Select the comment first and open its context menu with the right mouse button. Then use the menu item "Delete comment" to delete the comment.

Deleting a comment with its context menu



Changing the comment-text

Change the text in the worksheet

The text of a comment can be changed directly on the worksheet. Click onto the text and edit it afterwards. When you stop editing (e.g. by selecting another element of the worksheet) the changes will be stored in the comment.

Editing a comment in the worksheet



Changing the text property

Alternatively you can change the comment text by editing the property "Text". Since this property always contains a constant string, no formula is accepted for this property.

Computing the comment text

Instead of displaying a constant text you can use also use a formula for the computation of comment text. This formula must be stored in the property "Formula" of the comment.

The computed text of a comment (defined with the property Formula) is used instead of the constant text defined with the property "Text".

Deactivating the export of comments

The properties Copy To Result and Export determine whether a comment will be transferred into a Snapshot (Copy To Result) or PDF or Excel output (Export). Set the appropriate property to "false" to prevent a comment from being transferred.

The context menu for comments

Comments offer, as mentioned above, a context menu. This menu only allows to delete an existing comment.

CHAPTER 4:

Using the Config-Editor

Contents of this chapter:

Starting the Config-Editor	134
Structure of the Config-Editor	136
Editing configurations	143

Starting the Config-Editor

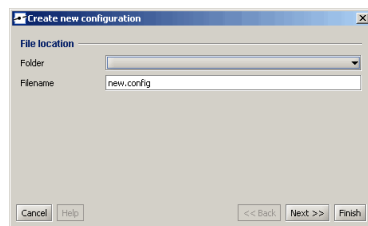
Creating new configurations

There are different ways create a new configuration within the Workbench: By using the main menu, the tool bar of the Workbench or by using the context menu of a directory within the Repository-Explorer.

Using the main menu or the tool bar

Use the menu item "Configuration/New Configuration" to create a new configuration. Then a wizard opens where you can define several settings for your new configuration.

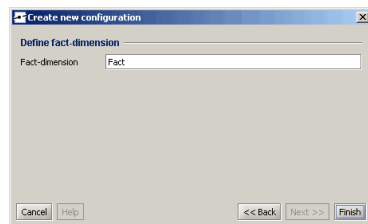
Choosing the filename



On the first page of the wizard you must define the filename of the new configuration and the folder where it will located. The filename of a configuration is very important because it will also determine the name of the model created by this configuration.

After you set the folder and the filename press the "Next >>" button to reach the next page of the wizard.

Choosing the fact dimension name



In the second and last page of the wizard you can define the name of the fact-dimension which will be automatically created by this wizard. Depending on your language there will be a default-name for this dimension but you can change the name if you want. The name of the fact-dimension must be a regular name and must not contains whitespaces or special characters.

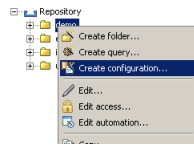
After you defined the name of the fact-dimension, use the button "Finish" to create the new configuration. Then the configuration editor will open, showing an empty configuration file.

After you created a new configuration, it will not be written to its folder until you save it.

Using the context menu of a folder

Using the context-menu of a folder to create a configuration is similar to use the Workbench-Menu or toolbar but the folder in the wizard will be pre-filled with the folder on which you opened the context-menu.

Open the context-menu of a folder and use the menu item "Create configuration..." to open the configuration wizard.



Opening existing configurations

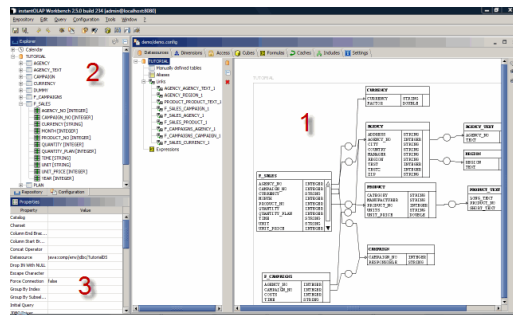
To open and edit an existing configuration you can double-click onto it within the Repository-Explorer or drag the file onto the empty space of your Workbench desktop.

Then the editor will open, showing the existing configuration with all its datasources, dimensions and other elements.

Structure of the Config-Editor

After creating or editing a new or existing configuration, the editor opens within the instantOLAP Workbench.

The editor



The editor is divided into three different areas:

1. The **Worksheet** representing the configuration itself,
2. the **Config-Explorer**, with which you can investigate datasources used by the configuration and use their tables, columns etc. for the configuration with drag&drop,
3. the **Property-Editor**, which is used to view edit the properties of all elements of a configuration

Also, there is a build-in **SQL-query tool** which allows to send free SQL statements to the databases when working with configurations.

The Worksheet

The Worksheet is the central element in the Configuration Editor. Here you can add datasources, dimensions, cubes, or caches to your configuration or edit them.

Tabulator bar at the top of the Worksheet



The Worksheet is divided into several parts, each of them can be reached by its tabulator at the top if the Worksheet:

- The **datasource editor** allows to add datasources (SQL databases) to the configuration and to add additional informations to them, like links between tables. Additionally, there is an ERM diagram which allows to visualize database-tables and to edit the relationships between them.
- The **dimension editor** is used to add dimensions to your configuration and to define, how they are loaded from the several datasources.
- The **access editor** allows to restrict access to dimensions or keys with rules.
- The **cube editor** is used to add cubes to your configuration and to define how they will load their data from datasources and to which dimensions they are connected to.
- In the **cache editor** you can add caches to your configuration in order to improve its execution time.

- In the **formula editor** you can define formulas combinations of facts and dimensions which can not be loaded from any datasource and have to be calculated therefore.
- In the **include panel** you can include other configuration-files and use the dimensions defined in the other configuration in your model.
- The **settings panel** allows to define some general settings for the model.
- The **notes panel** allows to write notes into the configuration file. These notes are stored together with the configuration and have no influence to the model at the runtime.

Each of this editors, except the notes panel, is described in the section "Editing configurations". To open one of these editors or panels simply click onto the tabulator at the bottom of the Worksheet.

Some editors need the dimensions defined in the dimension-editor to work, therefore changing to these editors can force the Worksheet to build a temporary model containing all dimensions of the configuration (without its cubes, caches or formulas). Whenever the editor is loading the temporary model, a small message-popup saying "Loading model ..." will appear in the lower right edge of your Workbench. Then you will have to wait until the dimensions are loaded.

If an error occurs while loading the model, the panel you switched to will not open until you fix the error in the dimension-definition. Then change back to the dimension-editor and fix the error.

The Configuration explorer

The Configuration-Explorer is located at the upper left corner of the Workbench. With this explorer you can investigate the datasources used by the configuration with all their tables, columns and other elements offered by the data sources.

Also, the explorer is the most important source for developing configuration using drag&drop. All elements of the databases can be dragged out of the explorer into the configuration with the mouse. You can drag elements onto the worksheet or onto the properties, which are displayed in the Property-Editor.

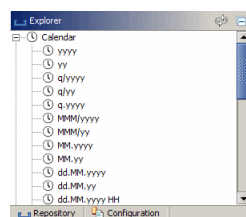
The Explorer is displayed as a tree-view (comparable with the File-Explorer from Microsoft Windows). Some elements can contain sub-elements, which can be opened (and closed) by clicking on the plus symbol to the left of the elements.

The Model-Explorer uses the same location as the Repository-Explorers. You can switch back and forth between the explorer with the tabulators below the panel.

Exploring the calendar

The calendar is always the topmost element in the Explorer. It offers the most frequently used date patterns which you can use when you create a time dimension.

The calendar in the Explorer



The calendar is no real datasource, it always exists in the Explorer and no datasource has to be connected to see it.

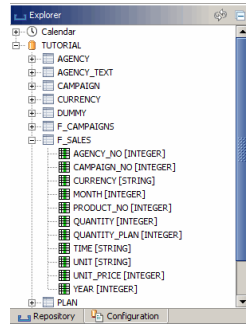
Exploring databases

In the explorer you will also see all databases for which connections in your configuration exist.

If you open a database in the explorer, all of its tables (or a subset of the tables, if you defined a table filter) will be displayed as sub elements.

If you open a table, you will see all columns of the tables and all its table expressions (which are "virtual columns" defined in the configuration).

A database in the Explorer



The Property-Editor

The Property-Editor is the table in the left, lower corner of the Workbench. The editor is used for investigating and editing the properties of any selected element. Whenever you select an element in the worksheet by clicking on it with the mouse, its properties are displayed and can be edited.

The table showing the properties

Property	Value
Cron Pattern	
Default Text Attrib...	Text
Description	
Level Names	
Load Children (fo...	true
Max Size	1000000
Name	Product
Storage	persistent
Visible	true

Editing properties

In order to change a property, you must click into the value field in the second column, right to the name of the property. After the cursor appears in the table cell, you can edit the value.

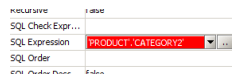
Changing properties in the table



Press the "Enter" key or click anywhere outside the cell to confirm the input. Then the system will check the new property value (e.g. for syntax errors in formulas). If an error was detected, it will be displayed as error-notification the Workbench.

If no error was detected, the editor will stop the input mode and you will be able to edit other elements afterwards. Otherwise, the input cell will show a red background.

An input containing an error



In the case of an incorrect input you must correct this before you can leave the input field. Alternatively you can use also the "ESC" to reset the property to the previous value and stop editing.

When selecting an element or after changing a property, the system checks all properties of the element and shows an error message if any error was detected.

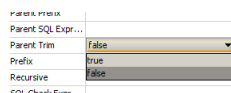
An error message produced by the property checker



Default values

Some properties offer default values from which you one by opening the combo box (the small triangle beside the input field).

Default values for a property

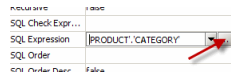


The previous value of the properties is replaced by clicking on a value. After the selection you still can change the property back to its previous value by using of the "ESC" key.

Special editors

For some properties (for expressions, SQL expressions, or cron-pattern) the Property-Editor offers special editors, which you can use to edit a value instead of using the simple input-field. If a property offers such an editor, a button containing three points "..." will be visible right to the input field. Press this button in order to edit the property with the editor.

The edit button for a properties

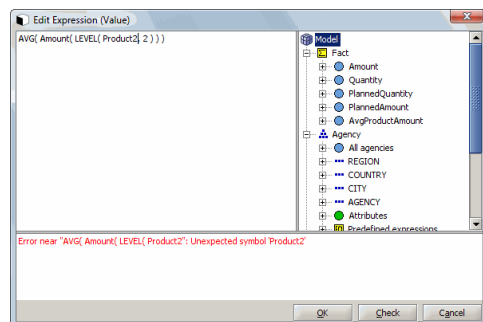


Also, most editors offer a check function, which you can use to examine your input without leaving the editor. Press the button "Check" to examine your input.

The Expression-Editor

With this editor you can create and edit formulas, which are used in many properties of the report-elements. Read the "instantOLAP Reference" for a complete overview of the instantOLAP expression-language and its functions.

The Expression-Editor



The editor is divided into three different areas: The input-area in the upper left area (for editing the text of the formula), the function-table on the right side and the message field at the lower edge.

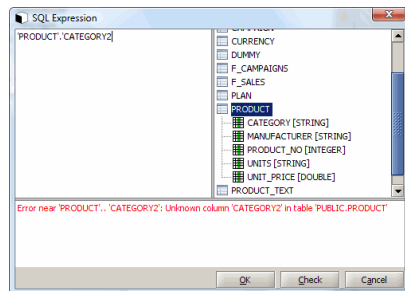
- In the function-table, all functions offered by the instantOLAP server (grouped by their topic) are listed. For each function you can see its syntax (i.e. the expected data types of their arguments). With a double-click on a function, the function will be insert at the current caret-position inside the input-field.
- The message field shows possible errors being detected in the current formula.

Use the "Check" button in order to examine your formula on its correctness and to, if any error was found, show the error in the message-field.

The SQL-Expression-Editor

With the SQL-Expression-Editor you can edit SQL-expressions which are needed in every SQL based element, e.g. for loading dimensions or facts out of databases.

The SQL-Expression-Editor



The editor is divided into three different areas: The input-area in the upper left area (for editing the text of the expression), the database-explorer on the right side and the message field at the lower edge.

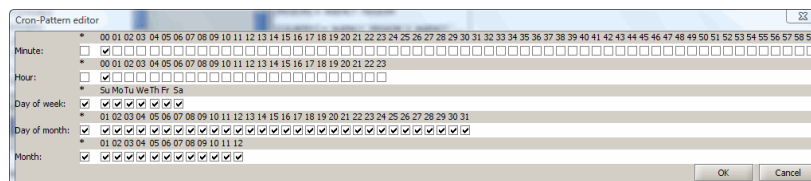
- In the database-explorer all tables and columns of the used database and some standard SQL-functions can be explored and added (by double-clicking the with the mouse) to your expression.
- The message field shows possible errors being detected in the current formula.

Use the "Check" button in order to examine your expression on its correctness and to, if any error was found, show the error in the message-field.

The Cron-Pattern-Editor

The Cron-Pattern-Editor allows to edit the cron patterns instantOLAP uses to define frequency repeated tasks like reloading dimensions or rebuilding stores.

The Cron-Pattern-Editor



The rows of the editor stand for the minute, hour, day of week, day of month and year in which the cron will be triggered. Each row must have a least one selected field, otherwise the cron-pattern will never be triggered. Instead of selecting each field of a row (e.g. if you want to cron-pattern to be triggered each hour) you can also check the "*" field at the beginning of the row.

The toolbar

Some of the buttons of the main toolbar of the Workbench are useful when working on a configuration:

Save



The "Save" button saves the current version of the configuration. Any previous version of the configuration will be overwritten and the productive version of the model will be replaced with the new version.

Save as...



The button "Save as..." allows to save the configuration under a new filename. After using the button, a file dialog will open and ask you for the new filename.

Undo



This button reverts the last change in the configuration.

Redo



This button reverts the last undo operation.

The SQL-query tool

When working with the source databases, the database explorer offers a useful SQL query tool. This tool allows to investigate the content of their tables or to perform free SQL statements on them.

The SQL query tool

AGENCY_NO	TIME	CAMPAIGN...	PRODUCT_NO	QUANTITY	UNIT	QUANTITY...	UNET_PR
2	012004	0	1	64	ST	69	1298
2	012004	0	2	77	ST	63	1421
2	012004	0	7	71	ST	68	1943
2	012004	0	12	49	ST	56	886
2	012004	0	15	44	ST	36	1505
2	012004	0	17	61	ST	69	1234
2	012004	0	20	43	ST	36	1750
2	012004	0	24	46	ST	49	1079
2	012004	0	25	85	ST	53	1685
2	012004	0	31	42	ST	38	1594
2	012004	0	32	57	ST	54	1361
2	012004	0	38	49	ST	45	1207
2	012004	0	40	67	ST	77	1644

There are two ways to open the query tool: You can open the context menu on a database in the database explorer or you can open the context menu on a single table.

The context menu for a database contains an item "Execute SQL query...". This will open an empty query window without any SQL query.

The context menu for a table contains an item "Show table-data". This also opens the query toolbar, but with a statement "SELECT * FROM <table>" which will instantly be executed. This statement will show the content of the table with all of its columns.

Elements of the query tool

The query tool contains three elements: The large text box where you can enter and edit the SQL statement, the result window below the text box and a toolbar.

The result window will show the result or, if any error occurred while executing the query, error messages.

The toolbar

The query tool contains a small toolbar at the right border. It offers two buttons:

Execute



This button executes the SQL statement or, if a part of the statement is selected, the current selection of the statement. By using selections you can hold multiple statements in the editor window and execute a single statement by selecting it.

Instead of using this button you can also press ENTER while holding the CTRL key.

*Change maximum
result size*



Any result displayed in the result window below the statement is limited to the first 1000 rows by default. You can change this limit with this button.

Editing configurations

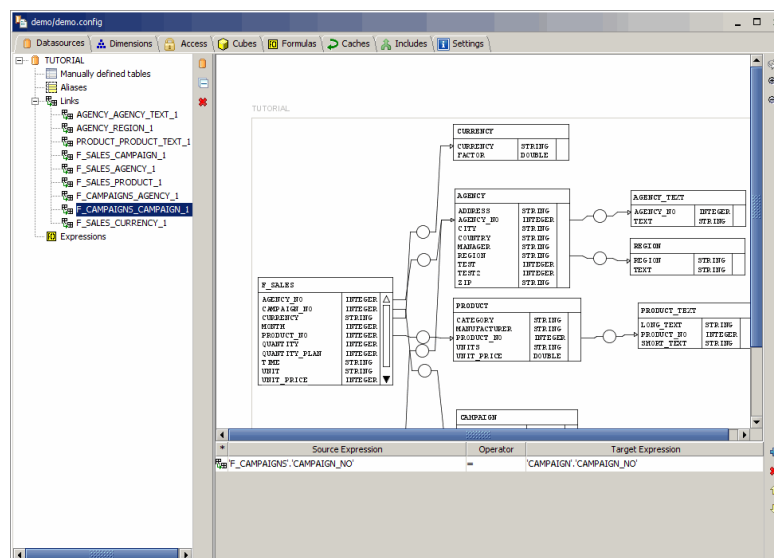
Editing datasources

Opening the datasource panel

Before you can create new or edit existing datasources, you must open the datasource panel in the editor. Click onto the "datasource" tab to open the panel.

The datasource panel is divided into two main areas: The datasource tree on the left side and the ERM panel on the right side. Both elements contain a small toolbar on their right border.

The datasource editor



Nearly all elements in the tree and in the ERM diagram can be selected by clicking onto them. After an element was selected, all of its properties will be displayed in the Property editor.

Some element (e.g. links and table-definitions) also contain some sub-elements. They will be displayed in a small table below the ERM diagram. In the previous screenshot, the table shows the column-pairs for a selected link.

The Datasource tree

The datasource tree shows **all** defined datasources of the configuration. When opening a datasource in the tree, you can also see all additionally defined elements for a database:

- Manually defined tables
- Aliases
- Links
- Expressions

These elements are just simple grouping nodes and have no properties. They cannot be edited or deleted at all. All elements, their function, and how to edit them, are described later in this chapter.

The toolbar

The tree also contains a small toolbar which is primary used to create new datasources and to delete elements:

Add database



This button adds a new SQL datasource to your configuration. Before the datasource is added to the tree, a datasource wizard will open and ask you for the most important connection parameter.

Collapse all



The button "Collapse all" collapses the whole tree and all of its elements.

Delete selected elements



Use this button to delete any datasource or element from the datasource tree (except the grouping nodes). No element will be deleted unless you confirm the following dialog.

Datasource elements

In the tree, each datasource contains a number of nested elements. The following section describes the additional elements for databases:

Manually defined tables (and columns)

Usually instantOLAP will gather all table names and their columns after you created a datasource connection. The tables and their columns are visible in the database explorer left to the configuration editor.

Some databases or JDBC drivers are not able to list their columns. In this case you'll have to define the tables and columns manually, otherwise you cannot use them later in the configuration. The defined names must exactly match the real names.

However, this is very seldom and nearly every JDBC driver is able to list the names of the tables.

If you need to add a manual table definition, use the context menu on the item "Manually defined tables" nested under the database node. Use the menu item "Add table definition" here - then the system will ask for name of the table and create a new table node.

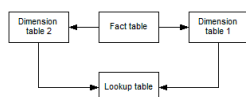
After you created the table you'll have to add its column names. Click onto the table node and use the table below the ERM diagram to add or delete column definitions to the table.

Aliases

Like in "pure" SQL, you must use aliases if you want to add a table more than once to a SQL statement. Especially when joining or filtering tables more than once, the usage of aliases is mandatory.

Consider the following example, where a fact table is linked to two different dimension table and both dimension table are linked to the same lookup table (which e.g. contains translations).

Lookup table used twice



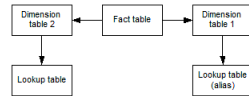
This configuration would return wrong and too few results, because the "Lookup table" is used and joined twice in the generated statement. The generator would create a statement like:

```
SELECT ... FROM FACT, DIM1, DIM2, LOOKUP WHERE FACT.ID1 = DIM1.ID AND
FACT.ID2 = DIM2.ID AND DIM1.LID = LOOKUP.ID AND DIM2.LID = LOOKUP.ID
```

So, the lookup table will be limited **at the same time** by dimension 1 and dimension 2 and the result will be empty or miss of the most result rows.

To correct this, you must create an alias on the lookup table use this for the second link:

Lookup table used as alias



Now, the lookup table is doubles and each instance (the original lookup table and the alias) are both limited only once. The statement would look like:

```
SELECT ... FROM FACT, DIM1, DIM2, LOOKUP LOOKUP1, LOOKUP LOOKUP2 WHERE
FACT.ID1 = DIM1.ID AND FACT.ID2 = DIM2.ID AND DIM1.LID = LOOKUP1.ID
AND DIM2.LID = LOOKUP2.ID
```

In instantOLAP, aliases only define their name, their original table and a optional WHERE condition. You can create as many aliases for each table as you need.

Links

Links describe to relationship between tables or aliases and are used by the SQL generator to add the JOINS to the SQL statements.

Whenever you use more than one table (e.g. in a cube), the generator will add them to the generated statement and will then try to connect them using the configured links of the datasource. If the generator is not able to link **every** table of the statement, it will raise an error message and a query will be interrupted.

Each link has a couple of properties:

- Links have a **logical name**, which is only used to display and edit it in the configuration editor.
- Links have a **source table**, a **target table** and a **direction**. The default direction is "LEFT" (from source to target), but it can also be "RIGHT" or "BOTH". The direction is very important for the generator, because it can limit the usage of link and can avoid the combination of tables which should never used together in a single statement (see below).
- Links can be INNER- or OUTER-joins. The type of the join influences the result of the SQL statements, because an INNER join only produces result-rows for primary / foreign key pair that are contain in the source **and** target table.
- Links contain an arbitrary number of column pairs, always on column from the source and one from the target table.
- Links have a WHERE condition, which is automatically appended to the generated statement whenever a link is used.

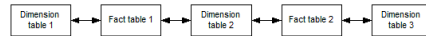
When creating and editing links it is very important to know the rules instantOLAP uses when assembling SQL statements:

1. instantOLAP collects **all** tables needed to load keys or to gather data from a cube.

- The generator tries to find at least one combination of links in which all of the used tables are joined with at least one other table. There must be one central table (usually the fact table or the basic dimension table), from which all other tables are reachable considering the **direction** of the links. Also, there must be no cluster of tables which is not linked to the rest of the tables.
- If the used tables cannot be linked, instantOLAP will add additional tables to the statement as "bridges" if there are links defined between the used tables and the new tables. It will use as few additional tables as possible.

Because the direction of the links is very important, it is suggested to

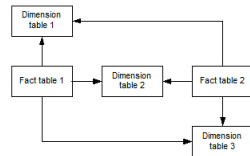
A bad link configuration



This link configuration would create wrong result, because the "dimension 3" is only reachable from the "fact table 1" through "dimension 2" and "fact table 2" (which would automatically added to the statement), and this permutes all rows from fact table 1 and 2 when using "fact table 1" with "dimension 3".

This happens because the link directions of all links is "BOTH", which allows the generator to use all links every time, even if this causes the include of other fact tables.

A good link configuration



This is a better and error-free link configuration: All links are directed **from the fact tables away** and all needed dimension tables are linked to all fact tables directly. Now, there is no need and possibility for the SQL generator to add fact tables to the statement.

Table expressions

A table expression is a "virtual" column which can be added to a table and basically is a SQL expression - whenever the virtual column it is used in a statement, the generator will insert its SQL expression instead.

Each table-expression has some basic properties:

- Each expression belongs to a single **table**. The name of the table is stored in the "table" property of the expression.
- Every expression contains a **name**. It will appear under its bound table as normal column with this name.
- The SQL expression defines the SQL which will be inserted in the generated statement whenever the expression is used.
- It has a "type" which describes the return type of the SQL expression.
- It contains a optional WHERE condition, which is automatically appended to generated SQL statement whenever the table-expression is used.

You can use table expression to simplify you configuration, because complex SQL expression can be reused by defining them as table expression.

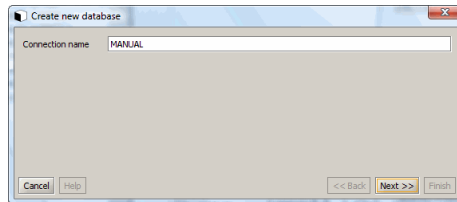
But they are also very useful to define a return type for SQL expressions, because whenever you use database dependent SQL code in your configuration, instantOLAP can have problems

identifying the return type and this can cause problems. Then, create an table-expression, set the proper return type and use it in your dimensions or cubes.

Creating new datasources

Use the "Add datasource" button from the toolbar to add a new datasource definition to the configuration. This will open the datasource wizard:

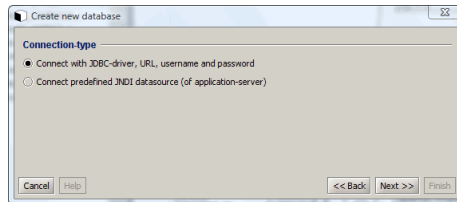
The datasource wizard



On the first page you must provide the **logical** name of the datasource. This is only the name under which the datasource will be referenced inside the configuration. The name does not need to match the real database name.

After you've chosen the datasource name, use the "Next >>" button to open the next page of the wizard.

Chose the connection type



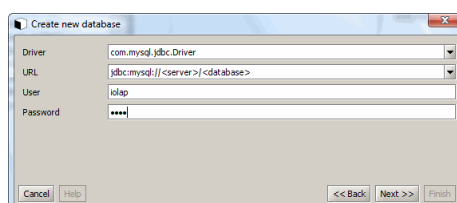
This page asks you for type of the database connection you want to use. There are two different ways to connect a database:

- **Connecting a database using a JDBC URL:** With this connection type, you'll have to provide all information like the used JDBC driver, the server URL, the username or password manually and all of this information are stored within the configuration.
- **Using a predefined JNDI datasource from the application server:** The administrator of the application server instantOLAP is installed on can provide predefined connections (datasources). In this case, you only have to tell instantOLAP the name of this connection (and sometimes the username and password, if they are not configured in the datasource).

The advantage of the predefined sources is that a configuration becomes less dependent on the database server, because its address is not hard-coded in the configuration. Especially when using development- and production-servers, the configuration can be copied from on to the other without changes.

In both cases, when using a predefined datasource or when establishing a manual configured connection, the JDBC driver which belongs to the database must be installed on the instantOLAP server.

Setting the connection properties

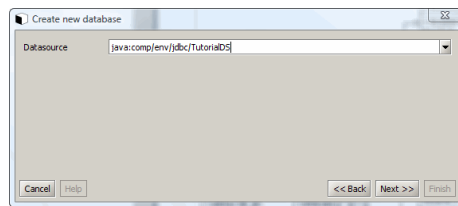


If you chose to use a **manual JDBC connection**, the next page in the wizard will ask you for the basic connection properties:

- **Driver:** Each JDBC driver has its unique Driver-path, which must be entered here. You find the path of each driver in the drivers documentation. For the most common drivers (Oracle, DB/2, Informix, Sybase, MySQL, MS-SQL etc) the driver paths are pre-configured and you can choose one of them by opening the drop down box.
- **URL:** This field expects a JDBC Connection-URL, which mainly contains the name of the used driver and the server URL. Usually the URL is "jdbc:<driver>:<url>", but the actual syntax is always dependent on the used JDBC driver. Like for the driver-paths, the URL syntax for the most common databases is also available as drop down list.
- **User:** The connection to the database will be established with this username.
- **Password:** The password which belongs to the user defined in the field before.

After you finished your entries, use the "Next >>" button to test the connection. If the server was able to connect the server without problems, the next page of the wizard will be displayed. If any error occurred, the message will be displayed and the wizard still shows this page.

Choosing a JNDI datasource

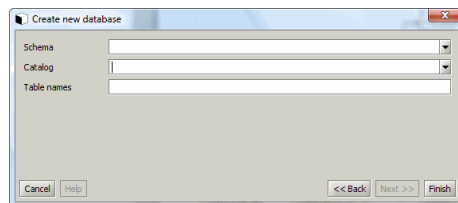


If you want to use a **predefined JNDI datasource** and used the second option in the last wizard page, a new page asking you for the JNDI path of the datasource will open.

Here must enter the complete JNDI path. Usually, the path starts with "java:comp/env/..." followed by the name of the datasource. Your administrator can tell you the complete path under which is located the source.

This wizard tries to load the list of available datasources from the server. But most application server will not list a datasource unless it is used at least once, so you might have to input the path manually.

Choosing catalog and schema



If both cases, when using a manual connection or a datasource, you can limit the connection to a single schema or catalog. The last page of the wizard lets you input or choose a schema- or catalog-name.

Not every database knows the concept of schemas or catalogs. If the database knows them, instantOLAP will query the database for all available schemas or catalogs and display them in the drop down boxes. Sometimes, a database has schemas or catalogs, but the driver is not able to list them - in this case you'll have to input the schema name or catalog manually.

The last field allows to input a pattern for table names if you want to limit the listed tables for databases. You can list a number of comma separated table names here and you can also use wildcards as "*" and "?" in the names.

Finally you can use the "Finish" button to create the datasource and to add it to the configuration.

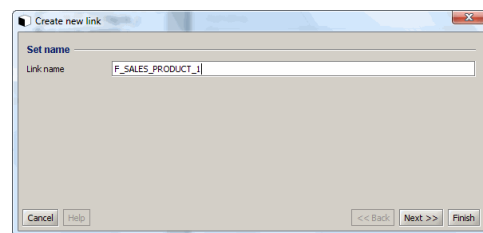
All settings you make in the wizard, like the connection properties, schema names, catalog names, table filters etc. can always be edited after the datasource was created by clicking onto the database element and changing its properties.

Adding a link to a datasource

There are two ways to add a link definition to a datasource: Adding it manually to the link node in the datasource-tree or creating a link with mouse in the ERM diagram (see below).

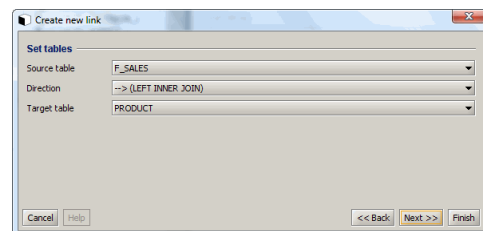
If you want to add a link manually, open the context menu of the "Links" node nested in the datasource and use the item "Add link...". After you clicked on the item, a dialog - the link wizard - will open.

The link wizard



The first page of the wizards ask you for the name of the new link. Each link has a unique and logical name which is only used for the visualization and navigation in the datasource tree.

The link tables and type



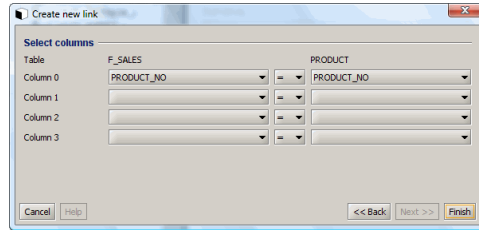
After you press "Next >>" you'll able to define the source and target table and the direction of the link. The source and target table can be any table of the current database. You can select both by using the dropdown boxes, they contain all tables of the database.

The direction of the link is initially a "LEFT INNER JOIN", which means the link points from the left table to the right and is a inner join. You can change the link to "LEFT", "RIGHT" and "FULL" in combination with "INNER" or "OUTER JOIN":

- An **inner join** links tables in that way that the result of the generated SQL statements will only contain records for primary / foreign keys that are contained in **both** tables. The direction for an inner join only determines if the SQL generator can use it when trying to find a link path from one table to another.
- An **outer join** creates also results for missing records in the source- or target-table. All selected columns of the missing record are filled with NULL then. For outer joins, the direction is important: In an "LEFT OUTER JOIN", the target record may miss, for a "RIGHT OUTER JOIN" the source record may miss and for an "FULL OUTER JOIN" both source or target records may miss.

Note that outer joins can generate much more CPU on the database server. Also, some database system do not know outer joins or have a limited number of outer joins that can be used in one statement.

The link columns



Editing links

After you created a link, you can edit it at any time. To edit a link, click onto the link in the database tree or in the ERM diagram.

Like for all other elements, the Property editor will show all properties of the link and you can edit them freely. But because the links also contain some sub-elements (the definition of the joined columns or expression from the source and target table), this elements will appear in a small table underneath the ERM diagram.

The columns of a link

	Source Expression	Operator	Target Expression
AGENCY.REGION		=	REGION.REGION

In this table, you can add new column pair to the link or edit the existing pairs. Each pair consists of three different columns:

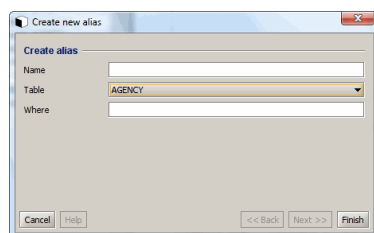
- The **source expression**, usually a simple column of the source table, defines the foreign key of the source table. You can use any expression here, but it **must** use the source table only.
- The **operator** defines the operator both expressions are compared with. The common operator is "=", but you can also use other operators like "<", "<=", ">", ">=", "<>", "IN" or "LIKE".
- The **target expression**, usually a simple column of the target table, defines the primary key of the target table. You can use any expression here, but it **must** use the target table only.

Use the "Add link" or "Delete link" buttons to add or remove pairs from the table. You can also arrange the order of the links using the "Move up" and "Move down" buttons. When deleting or moving a link, you must select it first by clicking on the icon in the first column of the table.

Adding an alias to a datasource

The only way to create a new alias is to use the context menu from the "Alias" node of the datasource. After using the menu item "Add alias", the alias wizard will open:

The alias wizard



This is a very wizard with only one page and three entries:

- **Name:** This is the name of the new alias. The alias will appear as a new table with this name in the database explorer after you created the alias.

- **Table:** This is the table the new alias is based on. This field is a dropdown box where you can choose the table from.
- **Where:** You can add an optional WHERE condition here, which always will be appended to the created SQL statement whenever the alias is used.

After you pressed "Finish", the alias will be created and added to the list of aliases.

Adding an expression to a datasource

In the datasource editor, three different context menus offer the creation of a new alias: The context menu of the "Alias" node under the datasource node, the menu of a table in the database explorer and the menu of a column in the database explorer.

Each item will immediately add an expression to the list of expressions, without any wizard. Depending on the menu you used, some of the expression properties are already pre-configured, the other are empty.

Click onto the alias under the "Alias" node to display the alias and to edit the missing properties.

Using the ERM editor

The ERM editor is a graphical representation of the tables of one or more databases. It is mainly used to visualize the tables, their columns and - most important - the relationship between the tables. Also, the diagram offers the most comfortable way to create links between the tables with drag&drop.

The diagram is initially empty and does not show any of the database tables. You must add the tables, or their aliases, to diagram manually. The added tables and their position in the diagram is stored together with the configuration file and restores automatically when loading the configuration at a later point of time.

The diagram will cluster all tables of the same database in a frame with the database-name below. It is possible (and necessary) to share the same ERM diagram for all databases.

Adding tables to the ERM

The only way to add a table or alias to the ERM diagram is to drag it out of the database explorer into the diagram panel. If the table did not exist in the diagram before, it will appear at the position you dropped it.

The tables are represented as boxes with the table title at the top and a maximum height. If the table has too many columns, a black scrollbar appears at the right border. You can scroll the columns and make other columns visible with the mouse wheel while the mouse is located over the columns.

A table with columns and scrollbar

F_SALES	
AGENCY_NO	INTEGER
CAMPAIGN_NO	INTEGER
CUSTOMER	STRING
MONTH	INTEGER
PRODUCT_NO	INTEGER
QUANTITY	INTEGER
QUANTITY_PLAN	INTEGER
TIME	STRING
UNIT	STRING
UNIT PRICE	INTEGER

Instead of tables you also add **aliases** to the diagram. Therefore a table can exist multiple times in the diagram, every time with a different alias (or as original table). It is very important to use aliases when defining the relationships between tables, otherwise you may create circular relations and the SQL generator would generate invalid statements.

Moving tables in the ERM

To move a table within the diagram, you can grab its title and drag the table freely within the ERM. The links will rearrange themselves while dragging the table.

Removing tables from the ERM

If you want to remove a table from the diagram, you must open its context menu (use the right mouse button anywhere on the table) and click onto the item "Remove from ERM".

Removing the table from diagram will neither delete the table from the database or the connected links from the datasource. It only removes the visualization, the table can be added to diagram to any time again.

Moving the viewport

To move the viewport and to show other parts of the diagram you can use the scrollbars at right and bottom border or you can click anywhere in the diagram and drag the visible viewport.

The toolbar

The refresh button



Use the "Refresh" button from the ERM toolbar to repaint all tables and all links.

The zoom buttons



The zoom buttons in the toolbar allow to increase or decrease the zoom of the diagram. If the zoom becomes too low, the text from the tables will disappear and only the frames of the tables and the links stay visible.

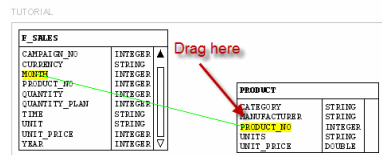
Instead of these buttons you can also use the mouse wheel to adjust the zoom while holding the mouse anywhere over the diagram except on a table.

Creating links in the ERM

Creating and managing links is the main purpose for the diagram. To create a link between two tables, locate the mouse on the columns of the source table which represents the foreign key (e.g. a column "ProductID" in a fact table).

Then hold the mouse button pressed and drag the mouse to the primary key of the target table (e.g. a column "ID" from a product table). While dragging, a line will indicate you are currently creating a link.

Creating a link in the ERM



A target column will be highlighted while you drag the mouse over it and if it would be a valid target. It wouldn't be valid, if it is from the same table than the source column or if both tables are already linked.

If the target column is valid and you release the mouse button, the same link wizard describe before will appear, but this time prefilled with all necessary information to create the link: A generated name, the source- and target-table, the link direction and the columns.

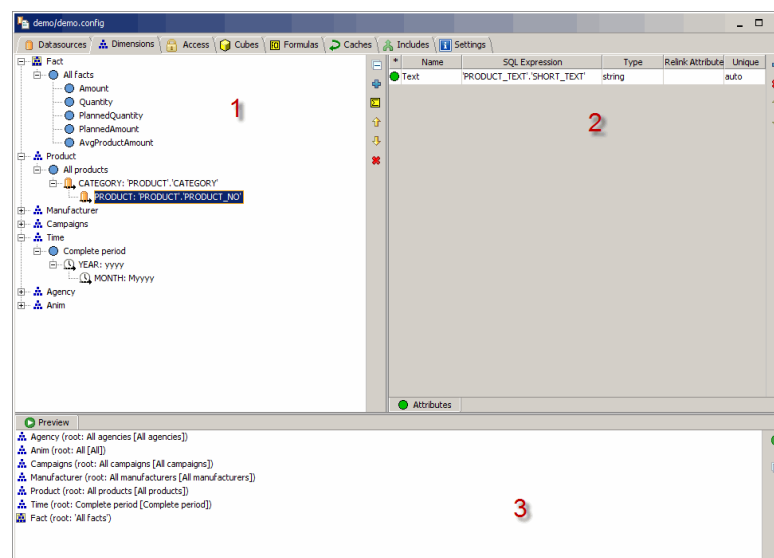
Only if your link should contain more than one pair of columns you must click through the wizard pages to the last one and add other column pairs to your new link.

Editing dimensions

Opening the dimension panel

To create new or edit existing dimensions, must open the dimension panel in the editor by clicking on the "dimension" tab in the configuration editor.

The dimension editor



There are three different regions in the editor:

1. The **dimension tree** allows to create new dimensions or to delete existing dimensions and to configure their Key-Loaders.
2. The **attribute table** at the right side allows to edit the attribute configuration for each loader.
3. The **preview** at the bottom can be used at any time to test and preview your current dimension configurations.

Editing dimensions

Creating a new dimension

Before you can edit a dimension, you must create it. Use the "Add dimension" button from the toolbar right to the dimension tree to create a new dimension.

Add dimension



A simple dialog will open and ask you for the name of the new dimension. If the name does not exist yet and is valid, the new dimension will appear at the end of the tree after closing the dialog.

The new dimension will only contain one constant key, the root-key for this dimension. It is mandatory and cannot be deleted, but you may change its IDs and all other of its property at any time.

Adding Key-Loaders to dimensions

After you created an empty dimension, you can add the Key-Loaders to it. A Key-Loader generates keys from different datasources (depending on the loader type) to the dimension whenever the dimension is build or synchronized.

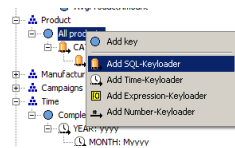
There are different types of Key-Loaders. All of this loader types are described in the "Concepts" chapter of this manual:

- **Constant keys** are "loaders" which add a single key, with a constant ID and constant attributes, to the dimension,
- **SQL-KeyLoaders** load keys from SQL databases by generating IDs defined by an SQL expression,
- **Time-KeyLoaders** generate keys using a date pattern,
- **Expression-KeyLoaders** generate keys defined by a instantOLAP expression and can use other dimensions as input,
- **Number-KeyLoaders** simply generate keys with numbers as IDs within a defined range

To add a new Key-Loader to a dimension, you can use the context menu of the dimension itself, the root key or any other loader or key of the dimension. The new Loader is nested automatically under the element whose context menu you used.

Depending on the type of the element, its menu offers all types of Key-Loaders or only the same type the element is of. This is, because SQL-KeyLoaders can only have other SQL-KeyLoader as children, Time-KeyLoaders only other Time-KeyLoaders and so on. Only dimensions and constant keys can embed any kind of loader.

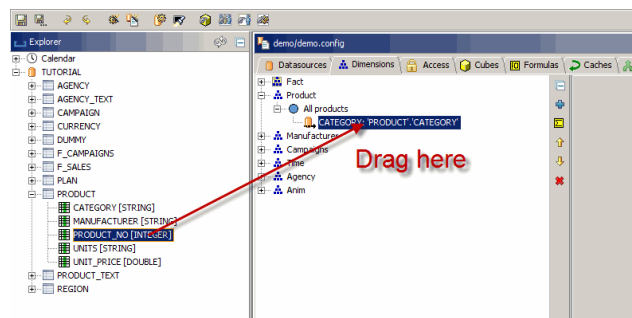
The context menu of an key



E.g., the context menu of the root-keys contains items to create all of five loader types. This is usually the place where you add the first loader for a dimension.

An easier way to create SQL-KeyLoaders or Time-KeyLoaders is to drag columns or time-patterns out of the datasource-explorer at the left border of the Workbench. This will automatically create a new loader with the column or pattern as ID-expression.

Creating a SQL-KeyLoader with drag&drop



For example, moving the column "PRODUCT.PRODUCT_NO" out of the explorer on the existing loader "PRODUCT.CATEGORY" would create a new Sql-KeyLoader with the column as expression and place it as new child under the category loader.

If you drag columns or patterns into the tree, the mouse cursor will indicate if the target position is valid or not.

Adding facts

Constant keys can only be created with the context menu of a dimension or other constant keys.

But in the case of creating a new fact (which is also only a constant key in the fact-dimension) you can use the "Add fact" button from the toolbar. This will append a new fact under the root-key of the fact-dimension.

Add fact



Copying Key-Loaders

The only way to copy an existing Key-Loader to another parent or to another dimension is to use drag&drop while holding the CTRL key pressed.

Moving Key-Loaders

The order of the Key-Loaders within the dimension also determine the order of the keys, because every loaded key is always appended as the last child under its parent.

The move buttons



So if you need to change the order of the keys and therefore of the Key-Loaders, select the loader and use the "Move up" or "Move down" buttons to arrange the loaders within their parents.

You can also move dimensions from one parent or even to another dimension using the mouse and drag&drop.

Changing the dimension order

The position of a dimension within the dimension tree can be important whenever you want to link dimensions with each other: Every attribute in a dimension can be a link to a key of another dimension, but this other dimension must be loaded **before** the attribute, otherwise it will not be visible as type in the attribute panel.

Move up, Move down



Use this buttons move the selected dimensions up or down in the tree. After you moved a dimension upwards, it will appear as attribute-type in all following dimension-attributes.

Deleting dimensions or Key-Loaders

Delete selected elements



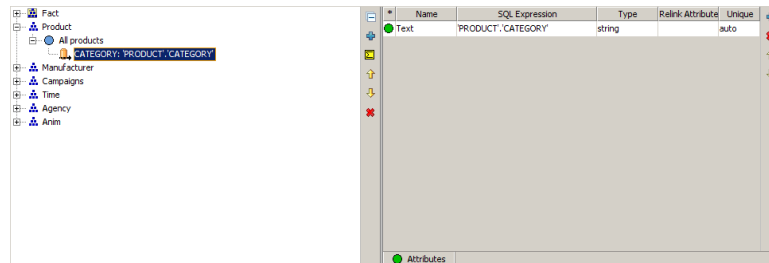
Use this button to delete selected dimension or Key-Loaders from the dimension tree. The Workbench will not delete the elements unless you confirmed the dialog which appears after using the button.

Adding attributes to loaders

Each Key-Loader (except Number-KeyLoaders) can contain an arbitrary number of attribute-loaders. This attribute loaders will add attributes to the keys while they are loaded from the datasource, calendar etc.

To edit the attribute loaders of a Key-Loader, select the loader in the dimension-tree. The table right to the tree then show all currently existing attributes of the loader (or is empty, if no attribute is defined yet).

The attribute table



You can add new attributes with the toolbar of the table. Use the "Add attribute" button to add a new row to the table.

Add attribute

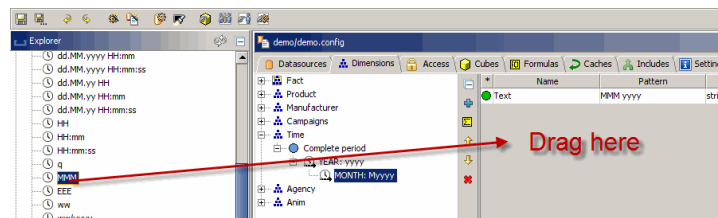


This will create a new and empty row within the table. Now you can edit the properties of the new attribute by clicking into the cells and changing their content. Some properties offer dropdown boxes or edit buttons (e.g. for SQL expression), which open a more comfortable editor.

Instead of editing an attribute within the table, you can also select it by clicking on the icon in the first row and display all of its properties in the property table. This will show all of the properties, in difference to the table which only displays the most important ones.

Like when creating Key-Loaders, you can also use drag&drop and drag columns or time-patterns out of the configuration explorer into the attribute table. Simply drag the column or pattern from the explorer into the free space of the table. You must select the loader before to display the correct target table.

Adding an attribute to a Time-KeyLoader with drag&drop



In this example, dragging the "MMM" pattern from the calendar adds a new attribute to the Time-KeyLoader, with the dragged pattern as generating pattern.

Removing attributes

To remove one or more attributes from a loader, select them by clicking on their icon in the first column and use the "Delete selected attributes" button.

Delete selected attributes



The attributes will be deleted without any prior confirmation dialog.

The preview

The preview panel is a tree at the bottom of the editor and can be used to test and investigate the dimension as they will be loaded with the current configuration of your dimensions.

The preview tree



Use the "Preview" button from the toolbar to load all new or changed dimensions and to display the result in the tree. If no error occurs, the root keys of all dimension will become visible, otherwise an error message will be displayed and the tree stays empty.

The preview button



To navigate inside a dimension, open it by double clicking on its root-keys. The key node will open, and you can see all keys of the first level. Each key can also be opened by double clicking it or by using the plus sign of its node.

For each key, the tree displays its ID and its default text (if existing) in brackets behind the ID. If a key contains attributes, this attributes will be displayed with their name and values before the children of a key. You can recognize attributes on the color of their node: Keys have blue icons and attributes have green icons.

The collapse button



Use the collapse button to fold all keys of the tree and to bring the tree into its original state.

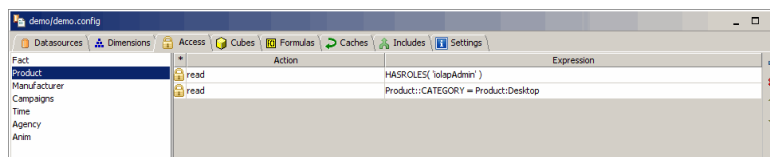
Limiting the access to dimensions

Opening the access panel

If you want to restrict the access to dimensions for certain users or groups, you must open the access editor - which is not part of the dimension editor. To open the editor, click on the "Access" tab in the configuration editor.

The access editor is a simple panel, with a list of all dimensions at the left and a list of access-rules on the right side.

The access editor



The table at the right side always display the access rules for the currently selected dimension in the left list.

Access rules

In instantOLAP, the access on dimension (or better: on keys)is controlled by a list of boolean expression. If **any** of these expression returns "true" for the current user and for the current key (the rules are performed on any single key the user wants to access), the user is able to use the key in his query.

Because the expression usually check the user and/or his roles and the current key, common functions used in access expressions are HASROLES and HASUSER. For example, a role "HASROLES({Product::CATEGORY})" would grant access to any product if the current user owns a role with the name of the product category.

If a dimension contains no single access rule, every user has access on any key, for reading and for writing.

Adding new rules to a dimension

Use the button "Add rule" from the toolbar to add a new and empty rule to the table.

Add rule



After you created the row, you can change the type of the rule from "read" to "write". Write rules are checked whenever a user tries to write data into a cube, all read access is controlled by the "read" rows.

The second and last column of the rule is the "Expression" columns which contains the boolean expression for this rule. You can edit the expression inside the table or use the edit button "..." in the cell to open the more comfortable expression editor.

Changing the rule order

If you want to change the order of the expression, select one or more rows by clicking on the icon in the first column and use the "Move up" or "Move down" button to move them inside the table.

Move up, Move down



The order of the rules has no effect on the access-restrictions, because a user has access on a dimension whenever at least one rule returns "true" - independent on its position inside the table. However, the order of the rules can have a small performance impact because rules matching quite often should be placed at the top (then the system has not to check the following rules).

Deleting rules

To delete a single or multiple rules from the table, simply select them and click onto the "Delete selected rules" button in the toolbar.

Delete selected rules



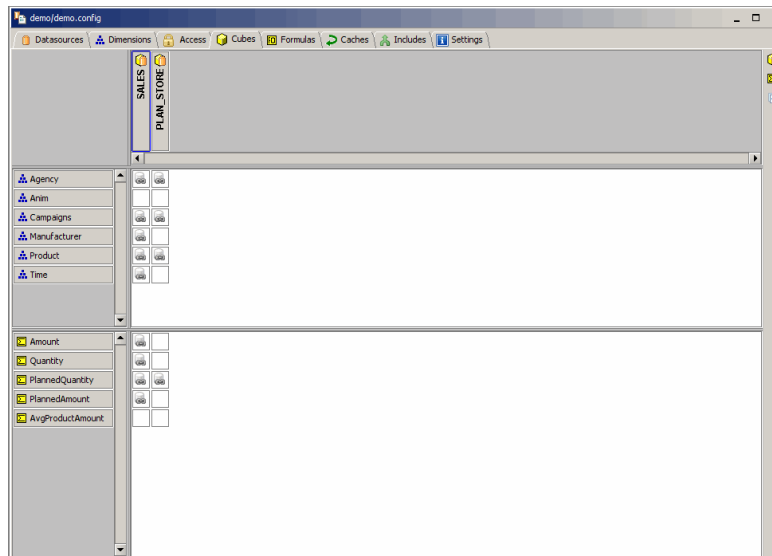
This will delete the selected rules without any prior confirmation.

Editing Cubes and Stores

Opening the Cubes panel

Click on the "Cubes" tabulator in the configuration editor to open the cube panel. If all dimensions are already loaded, this will immediately display the cube editor.

Otherwise, a "Loading..." message will be displayed until all dimensions are loaded. If no error occurs, the panel is displayed - otherwise an error message is shown and you must return to the dimension editor and correct the errors before you can edit the cubes.

The Cube editor

The editor shows all cubes and their dimension- and fact-mappings in a matrix: Every cube is displayed as a column at the top, all dimensions and facts are displayed as rows at the left. The main area of the matrix is split into two regions: The upper half for dimension mapping and the lower half for fact mappings.

The mapping symbol

Each cell can be empty or contain the mapping-symbol. This symbol shows if at least one mapping exists in the cube for the dimension or fact in that row. This gives you a short overview of all cubes and which facts they deliver for what dimensions.

Editing Cubes

Creating a new Cube

To create a new cube, use the "Add SQL-Cube" button from the toolbar at the right side. This will instantly add a new cube to the cube-matrix.

Add SQL-Cube

The new cube uses the first of all existing datasources by default. To use another datasource, you must change the "Database" property of the cube.

Changing the Cube properties

If you want to change the cube properties, you must select its header which represents the cube itself. Click on the header to select it - a blue border around the header indicates a valid selection.

A selected cube

After a cube was selected, its properties are displayed in the property editor and can be edited now.

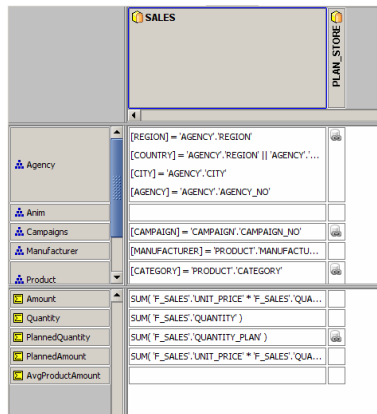
Unfolding and folding Cubes

Initially, all cubes are folded and their header and mappings are displayed as a small column. The cube name is shown vertically and existing mappings (if there is one or more mappings) are indicated with the mapping symbol in the matrix.

Before you can edit a cube, you must unfold it. You can use the context menu of the cube-header and use the "Expand" item. Or double-click on the header or on any mapping-cell (with or without mapping) to unfold the cube.

After the cube was unfolded, all mappings of the cube are displayed in the mapping cells. If there is more than one mapping in a cell, the row becomes higher and the mappings are show in a list.

An expanded cube



If you want to fold a cube, you can again use its context menu or double click on its header. To click on a mapping does not work here, because this would only select the mapping.

Collapse all



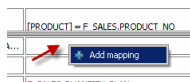
As an alternative, you can use the "Collapse all" button from the toolbar. This will collapse all cubes in the matrix.

Mapping facts to Cubes

After a cube was created and unfolded, you can map the facts and dimension for which the cube should deliver data.

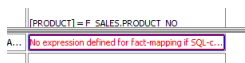
To map a fact, you must create a new mapping entry in the fact cell. Use the right mouse button over cell to open the context menu of the cell.

The fact mapping menu



For an empty cell, the menu only contains one item: "Add mapping". This items create a new mapping entry for the fact. Because the mapping is empty and does not contain any SQL expression (which is mandatory), it displays an error message at this point of time.

An empty fact mapping with error message



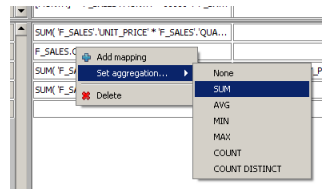
Now you can edit and correct the mapping. Click on the mapping cell to select it (if it is not selected yet) and to display its properties in the property editor. The most important and also mandatory property is "SQL Expression".

Usually, a fact is stored as a simple column in the fact table. In this case, the basic SQL-Expression would be "<fact-table>.<fact-column>" (e.g. "F_SALES.QUANTITY"). But a fact should always deliver **aggregated** values for the case it queried for a higher aggregation level of one of the dimension. Therefore, you need also to surround the expression with an aggregation function as SUM, MIN, MAX, AVG, COUNT or COUNT(DISTINCT).

Fact should always be mapped with an aggregation function (unless they are used in list-reports).

You can add the aggregation manually to the expression (e.g. "SUM(F_SALES.QUANTITY)") or use the context menu of the fact mapping after you entered the basic expression.

The default aggregation functions



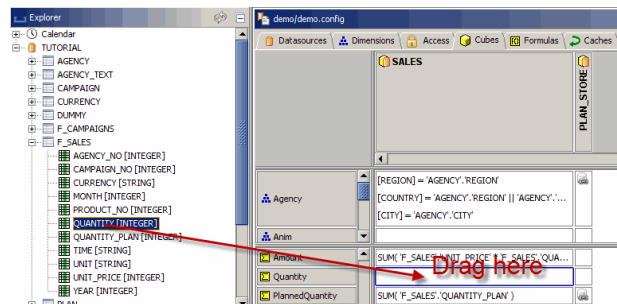
This menu offers the basic SQL aggregation functions mentioned above. Select one of them to change the expression and to add the function.

You can also use more complex expression for the fact or use other aggregation functions. For example, in the "TUTORIAL" cube, the fact "Amount" is mapped to the calculated expression "SUM(F_SALES.UNIT_PRICE * F_SALES.QUANTITY)".

Mapping facts using drag&drop

The faster and easier way to map facts in to use drag&drop. You can drag columns out of the database explorer and drop them in an empty fact-mapping cell of the cube.

Mapping facts with drag&drop



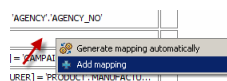
This create a simple mapping for this column. Then, you can add an aggregation using the context menu of the mapping or edit the expression in the property editor.

Mapping dimensions to Cubes

Like facts, dimensions can be bound to SQL expressions inside the cube. But in difference to facts, dimensions usually must be bound multiple times, e.g. for each level of the dimension (because different level usually need to be aggregated by other columns of the fact- or lookup-table).

Creating mapping entries for dimensions works similar as for facts. Use the context menu of an empty dimension cell and use the "Add mapping" item to create a new and entry.

The dimension mapping menu for empty cells



Then you can edit the item and enter the SQL-expression for the mapping. If the mapping is not valid for the whole dimension (which usually is not the case), you must limit it to a part of the dimension. The following limits are possible:

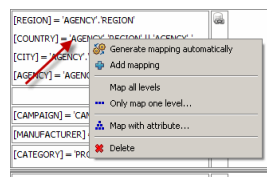
- You can limit a mapping to a specific **level** of the dimension.
- You can limit a mapping to keys containing a specific **attribute** and bind the result of the SQL-Expression to the attribute value instead of the key-IDs.
- You can limit a mapping to a **single key** of the dimension. In this case, no SQL-expression is necessary and only the optional WHERE clause of the binding is used.

The limitation to an attribute is usually used to bind keys with an incompatible id (e.g. product keys have the product name as id, and contain the product-ID as attribute, the product-id from the fact table should be bound to the attribute instead to the key itself).

It is also possible to combine limits and bind an SQL expression to keys with and specific attribute **and** a specific level.

The easiest way to limit a dimension mapping is to use the context menu of an existing mapping.

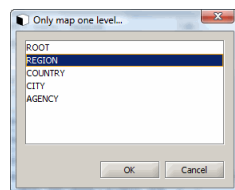
The context menu for existing dimension mappings



There are two different items to limit the mapping: "Only map one level..." to limit the binding to a specific level and "Map with attribute..." to map a specific attribute instead of the key id. Also, there is an item "Map all levels" which removes an existing level limitation.

When using the "Only map level..." item, a new dialog opens which offers all levels of the dimension for selection.

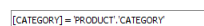
The level binding dialog



Select a level and press "OK" to limit the binding. Alternatively you could also edit the "Level" property of the binding in the property edit.

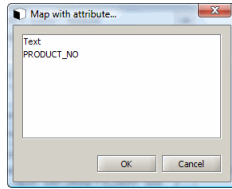
If a mapping is limited to a level, the matrix shows the level name in brackets and with a following "=" before the SQL expression.

A mapping with level



When using the "Map with attribute..." item, a similar dialog with the list of all unique attribute for the dimension opens:

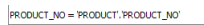
The attribute binding dialog



Select an attribute and press "OK" to limit the binding to a single attribute. You can also edit the binding properties to change or delete an attribute limitation.

After a mapping is limited to an attribute, the matrix will indicate this by write the attribute name with "=" before the SQL expression.

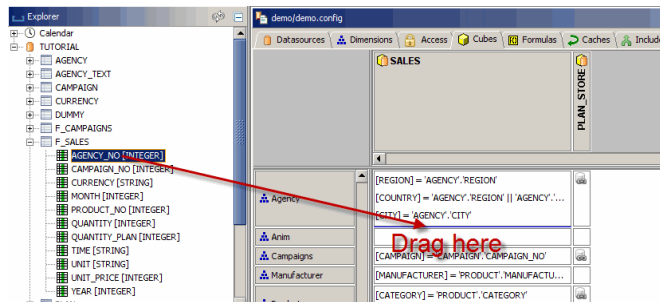
A mapping with attribute



Mapping dimensions with drag&drop

Also for dimension mappings, it is possible to drag columns out of the repository explorer and to drop them in dimension cells. This creates simple mappings for a single column.

Creating dimension mappings with drag&drop



When adding more than one mapping to a dimension cell, you must drop the column at the lower or upper border of the cell. A blue border will indicate valid drop positions when dragging a column.

After adding a dimension mapping using drag&drop, you can limit the mapping like any other mapping.

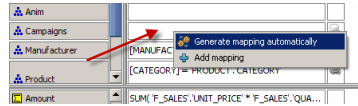
Automatic mapping

For simple dimension which are loaded with SQL-KeyLoaders, the workbench can automatically create the mapping for the cube. A dimension must fulfill the following criteria to be automatically mappable:

1. The dimension must be loaded with SQL-KeyLoaders **only** and every loader must load its data from the **same database** the cube does.
2. Every level of the dimension must own a level-name, because the automatic mapping **only** works on levels (only levels with names can be mapped in the cube editor).
3. There must not be more than one SQL-KeyLoader per level inside the dimension definition.

The automatic mapping uses all SQL-Expression from all SQL-KeyLoaders the dimension contains as mapping-expression. Each level from the dimension becomes one mapping in the cube.

The context menu for an empty dimension-cell



Open the context menu of the dimension mapping-cell and use the item "Generate mapping automatically..." to create the mapping. If the automatic mapping is possible, the workbench will delete all existing mappings for the dimension and add the new ones. Otherwise, it will raise an error message - e.g. if a level of the dimension has no name.

Note that the automatic mapping generates very simple mappings which are not necessarily the most performant solution. Especially the lowest level of dimensions are usually mappable directly within the fact-table, because the fact table references the lowest directly. When mapping the lowest directly, the system can use one table less in the generated SQL statement.

Changing the Cube order

The cube order can be very important and determines - in combination with the match modes of the cubes - where the system loads the data from:

- If the match mode is "first" or "exact", the first cube from the left containing a needed fact and mapping at least one (first) or exactly the number of needed dimensions (exact) will be used.
- If the match mode is "best", the first cube mapping the fact and the maximum possible number of dimensions will be used. If there is any other cube mapping the same number (but different) of dimensions, the left cube has a higher priority.

Because of this, you should control the order of the cubes and arrange them in for the case that the same facts are mapped in different cubes (if a fact is only mapped in a single cube, their order is without any meaning).

To arrange order you can use the context menus of the cubes or drag them with the mouse.

The context menu can be opened with the right mouse button while holding the mouse over the header of a cube. Depending on the cube position it contains a "Move right", "Move left" or both items. Use this items to shift a cube.

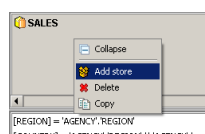
To arrange cubes with drag&drop, grab a cube header and move it to the left or right border of another header. A blue border indicates a valid drop position, if you release the button here the cube will be moved before or behind the target cube.

Deleting existing Cubes

To delete a cube you can use the context menu of a cube or the keyboard.

The context menu can be opened with the right mouse button while holding the mouse cursor of the header of the cube.

The cube context menu



It contains a "Delete" item - use this item to remove the cube from the configuration.

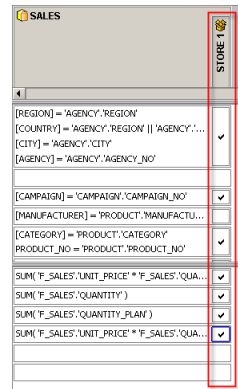
Alternatively you can select the cube with a mouse click on its header and use the DELETE key, to also delete the cube.

Editing Stores

Creating a new Store

To add a new store to an existing cube use the context menu of the cube and choose the item "Add store".

A store



Dimension	Mapping	Checked
[REGION]	'AGENCY','REGION'	<input type="checkbox"/>
[COUNTRY]	'AGENCY','REGION' 'AGENCY',...	<input checked="" type="checkbox"/>
[CITY]	'AGENCY','CITY'	<input checked="" type="checkbox"/>
[AGENCY]	'AGENCY','AGENCY_NO'	<input checked="" type="checkbox"/>
[CAMPAIGN]	'CAMPAIGN','CAMPAIGN_NO'	<input checked="" type="checkbox"/>
[MANUFACTURER]	'PRODUCT','MANUFACTU...	<input checked="" type="checkbox"/>
[CATEGORY]	'PRODUCT','CATEGORY'	<input checked="" type="checkbox"/>
PRODUCT_NO	'PRODUCT','PRODUCT_NO'	<input checked="" type="checkbox"/>
SUM('F_SALES','UNIT_PRICE' * 'F_SALES','QUA...		<input checked="" type="checkbox"/>
SUM('F_SALES','QUANTITY')		<input checked="" type="checkbox"/>
SUM('F_SALES','QUANTITY_PLA...')		<input checked="" type="checkbox"/>
SUM('F_SALES','UNIT_PRICE' * 'F_SALES','QUA...		<input checked="" type="checkbox"/>

Stores are displayed as an additional column nested under the cube. The name of the store is displayed in its header and for each dimension and fact mapped by the cube there is a checkbox right to the mapping.

Changing the Store properties

Like cubes, you can display and edit the store properties by clicking onto its header and selecting it.

The store properties allow to change the store name, its build properties or to mark it as not active. These are the most important properties:

- **Name:** The name of the store.
- **Cron Pattern:** This pattern defines the time schema when a store will be automatically rebuild. By default a store has no pattern and will not be rebuild unless it is triggered manually or a dimension has changed.
- **Build timeout:** The timeout for the store build process in seconds. The default value is 3600 (one hour).

Adding dimensions to a Store

After you created a new and empty store you will have to decide which dimensions you want the store to contain. Adding a dimension is very simple, just click onto the empty box next to the dimension mapping (in the column of that store) and a checkbox will appear and signal this dimension as being contained in the store. Alternative you can use the context menu of the empty checkbox, which only contains one item ("Add dimension to store").

After you added a dimension to a store, you can edit its settings for this dimension by selecting the checkbox with a single click. This will display the dimension properties, namely its name and the "Load Mode" of the dimension.

The "Load Mode" defines, if the hierarchy of the dimension will be calculated by aggregating the values bottom up or by querying the database for each single level. Read the "Concepts" chapter to learn more about the load mode. You can also change the mode by using the context menu of the checkbox.

The context menu of a store dimension



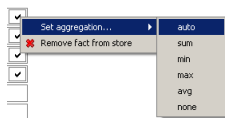
To remove a dimension from a store, you can either double click the checkbox or use the context menu of the checkbox and select the "Remove dimension from store" item.

Adding facts to a Store

Like dimension, facts can be easily added with a single click on the checkbox right to the fact mapping or by using the context menu of the empty checkbox.

To display the properties of the fact, select it with a single click onto the fact. The only property you may want to change is the "Aggregation" of the fact which can also be changed with the context menu of the fact.

The context menu of a store fact



The aggregation defines if the fact will be aggregated using the sum, average etc. value from lower dimension levels. The default aggregation for facts is "auto" which tries to automatically determine the aggregation from the bound SQL expression. If this is not possible or if you want to change the aggregation, you can change it from "auto" to one of the desired values "sum", "min", "max", "avg" or "none".

To remove the fact from the store, double click it (like dimensions) or use the context menu of the fact checkbox.

Changing the Store order

Like cubes, stores have a defined order with its owning cube and the order can be very important. Whenever the system tries to load data from a cube which contains more than one store, it queries the stores from left to right for the data. Therefore you may want to change the store order.

If your cube contains more than one store, you can rearrange them with drag&drop or by using the context menu of a store. Use the items "Move store left" or "Move store right" here.

The context menu of a store



Deleting existing Stores

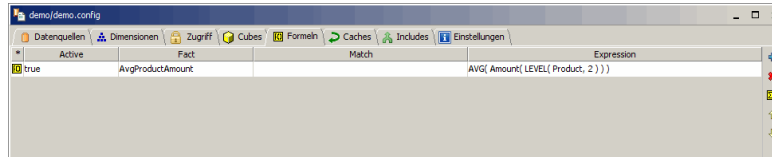
To delete a store you can select it and use the DELETE key or open its context menu and use the "Delete store" item.

Editing Formulas

Opening the Formula panel

Click on the "Formulas" tabulator to open the formula editor.

The formula editor



The editor contains a table with all existing formulas of the configuration. For each formula there is a number of properties displayed as columns in the table:

- **Active:** If this property is set to "false", the formulas will be removed from the model and not used. The default value is "true".
- **Fact:** Use this dropdown box to limit the formula to a single fact. This is normally useful because a formula usually calculates a specific facts and limiting a formula to a fact also increases the speed of a model. But you can also leave this empty entry and calculate every fact with this formula or use the match expression to limit it to more than one fact.
- **Match:** This boolean expression determines if a formula can be used to calculate a missing coordinate, e.g. by checking the current level of a dimension or anything else. For example, a match-expression "HASLEVEL(Product, 1)" would only calculate values for the level one of the product dimension. The match expression is optional.
- **Expression:** This mandatory property determines, how the formula calculates its results. The expression must return "values" (the result must be a sub-type of the type "Value").

Editing Formulas

Creating a new Formula

Add formula



Use the "Add formula" button to add a new formula to the end of the list. After you created a formula, you can assign it to a fact and change its match-expression and expression by clicking into the table.

Add fact



If you want to create a formula for a fact which does not exist yet, you can use the "shortcut" button "Add fact" to add a fact to the fact-dimension without opening the dimension editor.

Changing the Formula order

The order of the formulas can be very important, because their match-expressions could overlap and - in that case - the first formulas matching a coordinate will be used by the system.

Move up and Move down



Use the "Move up" and "Move down" buttons to arrange the formulas after selecting the by clicking on the formula icon in the first column.

Deleting Formulas

Delete formula



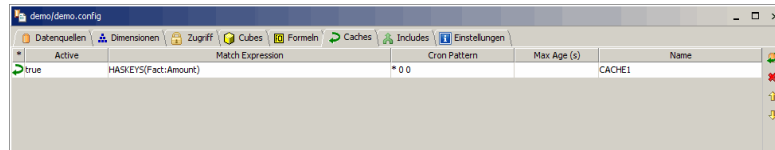
Select a formula by clicking on its icon and use the "Delete formula" button to delete it.

Editing caches

Opening the Cache panel

The "Cache" tabulator of the workbench editor contains a table with all configured File-Caches.

The cache editor



Active	Match Expression	Cron Pattern	Max Age (s)	Name
true	HASK/E/S(Fact:Amount)	* 0 0		CACHE1

For each cache, the table contains a number of columns with the following cache properties:

- **Active:** Only if this property is set to "true" (the default value), the cache will be used. You can disable a cache to test a model without a cache, e.g. when developing a cube.
- **Name:** The name of the cache. The name must be unique inside the model.
- **Match expression:** If this optional property is empty, the cache will store and deliver data for all coordinates. Otherwise, only coordinates for which this boolean expression returns "true" will be cached.
- **Cron pattern:** This cron pattern defines the frequency how often the cache will be synchronized and checked for outdated entries. You can use the edit button "..." next to the pattern to open the comfortable cron editor.
- **Max age:** This is the maximum age of cache entries. Whenever the system checks the cache (which is done as often as defined by the cron pattern), all entries older than this maximum age will be deleted from the cache. If this optional property is empty, no entry will outdate at any time.

Editing Caches

Creating a new Cache

Like in all other property tables, you can use the "Add cache" button from the toolbar at the right border to create a new row and cache.

Add cache



The new row contains a new cache with a generated name (which can be changed), no cron pattern, no maximum age and no match expression.

Changing the Cache order

Use the "Move up" and "Move down" button to change the order of the caches within the model.

Move up and Move down



The order of the caches can have an impact on the system, because the server tests the match expressions beginning with the first cache when loading or caching values. Therefore, if you have multiple caches and one cache without match-expression, this should be the last entry.

Deleting Caches

Delete cache



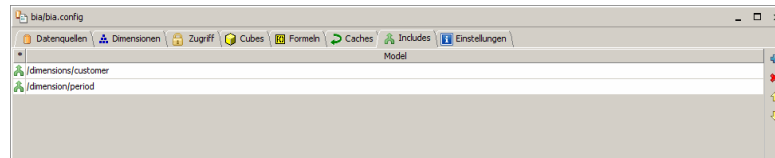
To delete a cache, select its row by clicking on the icon in the first row and use the "Delete cache" button.

Importing other models

Opening the Include panel

The include panel is a simple table with the paths of the other models which should be included in this configuration. To open the table, click on the "Include" tabulator.

The include editor



Editing Includes

Creating a new Include

Add include



Use the "Add include" button to add a new row to the table and enter the path of the other model into the new row. The path can be relative to the current model or absolute.

E.g a path "customer" would use a model "customer" from the same folder, a path "/dimensions/customer" from the folder "/dimensions".

Editing the Include properties

Existing includes can be edited by simply clicking into the row and changing the path.

Move up and Move down



You can also change the order of the including by using the "Move up" and "Move down" buttons, but this has no effect on the generated model because the dimension are finally sorted by their name inside the model.

Deleting Includes

Delete include



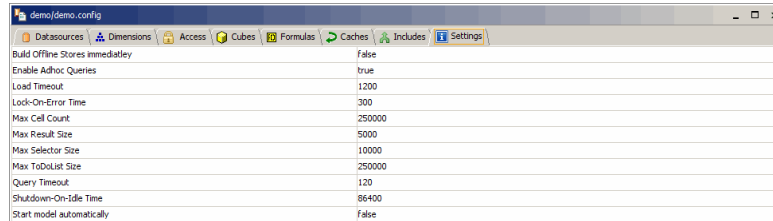
To delete an include, select a row by clicking on its icon in the first column and use the "Delete include" button from the toolbar.

Basic configuration settings

The "Settings" tabulator of the configuration editor allows to edit the basic settings for a configuration. This includes timeouts, limitations or the automatic startup for a model.

To open the setting, click on the last tabulator "Setting" in the configuration edit.

The model settings



Property	Value
Build Offline Stores immediately	False
Enable Adhoc Queries	true
Load Timeout	1200
Lock-On-Error Time	300
Max Cell Count	250000
Max Result Size	5000
Max Selector Size	10000
Max ToDoList Size	250000
Query Timeout	120
Shutdown-On-Idle Time	86400
Start model automatically	False

The settings panel is a simple table with a number of properties, which can be edited inside the table. The following settings are available:

- **Build stores immediately:** If this property is set to "true", all offline cubes of the model will be build immediately after the model is started and online. However, in the time span between the starting of a model and the successful build of a offline-cube, the cube may operate in the "adhoc" mode. The default setting for this property is "false".
- **Enable adhoc queries:** Since the 2.5 portal, users can perform adhoc-queries on models. But a model will **only** be visible for adhoc-queries if this property is set to "true". The default value is "false".
- **Load timeout:** This is the timeout for a model (in seconds) when it is initially loaded or synchronized. The default value is 1200 seconds.
- **Lock-on-error time:** Whenever the loading or synchronizing of a model fails, the usage of the model is disabled for the amount of time defined by this property (in seconds). This is, because a faulty model would, every time it is going to be used, try to load its dimensions and stores and unless it is fixed - users trying to use the model again and gain would cause a necessary CPU load on the server. The default value is 300 seconds.
- **Max cell count:** There is a maximum size for queries and their results, because queries cannot be infinite complex. This property defines the maximum number of **cells** (rows multiplied by columns) a result must contain. If the number of cells of a result exceeds this limit, an error will be raised. The default value is 250.000 cells.
- **Max result size:** Like the "Max cell count", this property limits the maximum result size - but, in this case, the maximum number of rows or columns a result must own. The default value is 5000 rows or cells.
- **Max selector size:** Selectors accept anything for their "options" property, but the number of options must not exceed the limit defined by this setting, otherwise the amount of data transferred from the server to the browser could become too large. The default maximum is 5000 options.
- **Max ToDoList size:** The ToDo-List is the list of all facts (or coordinates) a query needs to load from the model-datasources. Even if a query looks quite small and simple, it (and its formulas) could need a large amount of data to perform. Because the size of the ToDo-List is related to the memory and CPU- usage of the query, there is a default limit of 250.000 entries for it. If a model only contains offline cubes, no ToDo-List is generated and this property is not used.
- **Query timeout:** This is a very basic property - no query must take longer than the number of seconds determined by this property to perform, other a "timeout" exception will be raised and the query is interrupted. The default timeout is 120 seconds.
- **Shutdown-On-Idle time:** Every model being started and online used memory and CPU time (because the system is synchronizing its dimension and stores). Therefore, a model, which hasn't been used longer than the seconds defined by this property, will be shut down and removed from memory. The default shutdown time is 86400 seconds (one day).
- **Start model automatically:** Frequently used models should set this property to "true" as the first user performing queries on it does not has to wait for the model to start up.

After you edited the properties and saved the configuration, the changes will be used immediately.

CHAPTER 5:

Concepts

Contents of this chapter:

A quick overview	172
Models	173
Structure of a query	177
How instantOLAP works	191
Good model design	205

A quick overview

This chapter will give you a detailed overview about the concepts behind instantOLAP and how it works.

instantOLAP is a model driven OLAP system - this means all database related information is configured and encapsulated inside a model and the user of a model (which is basically the creator of a query or pivot table) does not need any knowledge about the underlying databases or about databases and SQL in general.

But even if you don't need to create new models and only will create queries based on existing ones, it is recommended to also read the first part of the model chapter in order to learn about important OLAP concepts like dimensions, facts and cubes and how they are realized in instantOLAP.

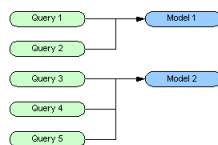
Models

Models

Reports within instantOLAP never directly access the datasources - they are based on a logical middle layer, the "models". A model is a logical abstraction of the data sources and represent their master data as "dimensions" and the values as "facts". These dimensions and facts determine the reporting possibilities for the queries.

Several queries can access the same model, but each query can only use one model.

Queries can share models



Elements

A model consists of the following elements, which are described in detail in the further text of this section:

- **Dimensions** with their **Keys**, **Hierarchies** and **Attributes** which represent the master data
- **Facts** (which are hold in a special fact-dimension) which represent the different values which can be used in queries
- **Cubes** as source for the values of the Facts. They represent the interface to the datasources and query the facts in realtime or preaggregate them in ahead - depending on their configuration.

Queries are only referencing the dimensions and facts of a model. All other elements (like cubes, formulas or caches) are invisible and only used for the internal processing of a model.

Dimensions, Keys and Hierarchies

What are dimensions?

In multidimensional data models intended for OLAP, data are viewed as points in a multidimensional space. Each dimension has a hierarchical structure, described by a directed graph of categories (levels), a set of members (keys) for each level and a parent / child relation between keys (which defines the hierarchy).

As an example, a dimension "Date" would e.g. contain the levels "Year", "Month" and "Day". The years, months and days would be then keys of this levels. Between the keys a parent / child relation would exist, e.g. each day would be a child of a specific month etc.

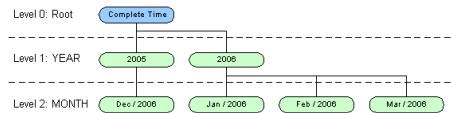
An important application of this structure is to use for it aggregation - the value of each can always be derived from its children keys.

Dimensions in instantOLAP

In instantOLAP, dimensions can contain an arbitrarily number of levels and keys. Only one hierarchy is allowed per dimension, therefore each key always has exactly one parent key.

The topmost level of a instantOLAP dimension is called the "root-level" and always contains exactly one key, the "root-key". This root key is mandatory and stands for the topmost aggregation possible for a dimension. Because this key is mandatory, each dimension in instantOLAP contains at least one key and one level. The root-key is the only key in a dimension without a parent key.

Levels and keys in an example time-dimension



Examples for a root key would be "Complete period" for a date dimension or "All products" for a product dimension.

Dimension as types

In the type-system of instantOLAP, each dimension becomes a single type in the type hierarchy and each key of a dimension is a value of this type. Each dimension-type is automatically a sub-type of the type "Key" in the type-system. Dimension-types are very important for links between dimensions (see below).

Keys

Keys are the members of a dimension in instantOLAP. Each key owns a unique ID within the dimension, a parent key (except for the root-key) and no, one or more children and a arbitrarily number of attributes.

The ID of a key is always a string (text) and *must* be unique within the dimension. Usually, the ID is defined by the loaders you create for a dimension when building the model.

Because each dimension builds its own type in the type-system of instantOLAP, each key is a value with the type of its dimension.

Levels

In instantOLAP, keys are organized in levels within an dimension. Each key is a member of exactly one level and each level (except the root level) can have a arbitrarily number of keys as members.

Levels have both an index and a name. The index of the first level (the root level) is always 0, the following levels have the indices 1, 2 and so on. The names of the levels are defined by the creator of a configuration. Level-names must be unique inside a dimension and they must be valid names (they must not start with numbers and not contain any special characters).

Attributes and Links

Each key can own a arbitrarily number of attributes. Attributes are additional informations for a key, e.g. a product-number for a product or an address for a customer. Each attribute has a name under which it can be addressed later in the queries.

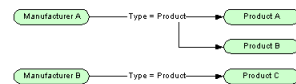
Attributes are defined on a dimension basis, each key of a dimension can contain no, one or more values for any of the attributes defined for the dimension. Not every key must contain

values for an attribute, therefore accessing an attribute of a key can also result to no values but won't raise an error in the type checker.

Because instantOLAP uses a strong type-system, attributes also use types. Each attribute is defined to be of a specific type and all values of this attributes may only contain values of this type, e.g. strings, integer-values or booleans.

A special case are attributes with a dimension-type, they may only contain keys from other dimensions as values. Because this attributes refer to elements from other dimensions, they act like "links" between dimensions. E.g. you could create a link between the two dimensions "Product" and "Manufacturer" with an attribute and use this link later in your queries.

*Links between
dimension keys*



Link have directions, they point from one key to another. Also, bidirectional links are possible and point from one key to another and backwards. In this case, each key has an own attribute (which can have different names) linking the other key.

Default text-attributes

Whenever Keys are used in queries, their IDs are displayed as text by default inside the table headers, selectors or any other place wherever they are used.

If the IDs are too technical (e.g. if you use database IDs or self generated IDs in order to make then unique inside the dimension), dimensions can define one of their attributes as "default text attribute" to be displayed instead of the IDs. This allows to display human-readable texts though the IDs are technical and unique. This default text attribute does not have to be unique, e.g. two product could have the same text though they own different IDs.

If a key contains no value of the default text attribute its ID will be displayed instead of the attribute in queries (like there was no default text attribute).

Facts

What are facts?

In OLAP systems, facts represent the business measures depending on the dimensions. Typical facts are "Turnaround" or "Price". All of these facts depend on at least one dimension - e.g. the "Turnaround" would depend on the Period, the Product, the Region and so on.

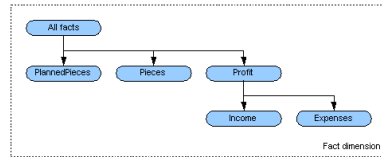
The fact-dimension

In instantOLAP all facts are defined as keys inside a special dimension, the so-called "Fact dimension" which exists in every model. Each fact is defined as a key in this dimension, together with some additional properties like the type of its values, its format, its unit or user-defined attributes.

The name of the fact dimension can be freely defined. Usually its name is "Fact".

Because facts are members of a dimension, you can define a hierarchy for them and also use this hierarchy in your queries. And because facts are simply keys defined in a dimension, you can make them selectable in selectors, use them in table headers, and use them in your queries and so on.

A example
fact-dimension



Types

The type of a fact determines which kind of data will be returned whenever a fact is used in expressions or queries. Usually fact return "Numbers" (Double or Integer values), but any other kind is also possible.

Formats

A fact can have a format which determines how numbers or dates will be formatted in the display of a query. If no format is specified, the standard format for numbers will be used. For other types than numbers and dates no format will be used and any defined format will be ignored.

Units

The unit of a fact is a small text being displayed in table headers to show the unit of a business fact. Typical units are "USD", "EUR", "Pieces" or "Days".

Cubes

OLAP's main representation of facts is via what is known as a *Cube*. A Cube is a multidimensional representation of facts that can be accessed quickly to provide specific information for any combination of dimension-keys (a tuple of keys from different dimensions is called a *coordinate* in instantOLAP).

For example a set of customers can be grouped by city, by district or by country. So with 50 cities, 8 districts and two countries there are three hierarchical levels with 60 members. These customers can be considered in relation to products. If there are 250 products with 20 categories, three families and three departments then there are 276 product keys. With just these two dimensions there are 16.560 (276 * 60) possible coordinates for which a cube can deliver data.

In instantOLAP a model can contain multiple cubes but will decide automatically which of the cubes must be used whenever a query accesses data. Therefore cubes are invisible to queries and model users.

Structure of a query

Before you create your own reports, you should get familiar with the structure of a query, the query elements and their functions. This section gives you a detailed overview of the structure and the elements of queries.

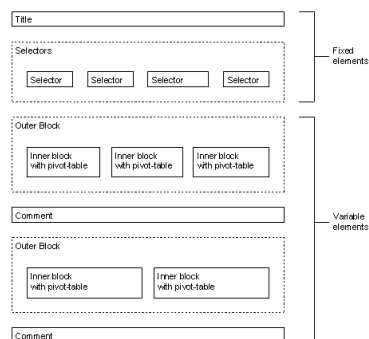
Elements of a query

Within a query there are fixed elements (which exactly occur once) and variable elements, which you can use as much as desired (or leave them out).

The fixed elements are the title and the selector area. These elements appear in each query, even if you did not specify any title or selector. The title and the selector are always placed in the topmost position.

The variable elements are the blocks (there are inner and outer blocks, which are encapsulated into each other) with pivot tables and comments. You can use as much as wanted of this elements and set their order freely.

Structure of a report



Inner blocks are always placed within outer blocks and pivot tables are always placed within inner blocks, so that the pivot tables of a report are organized in a table-like structure. On the highest level of a query, only outer blocks and comments are allowed.

Order of the elements

The order of the outer blocks and the comments can be defined freely. Also, you can define the order of the inner blocks within an outer block at will.

Differences for pivot-tables

Pivot-tables have the same structure as queries, but they always only contain one outer and one inner block. They are just versions of queries but, except for the limitation of blocks, have the same possibilities.

Selectors

Selectors are interactive elements giving the user the possibility to limit the query to certain elements of a dimension (keys).

There are different kinds of selectors, with which the user (depending on the type of the selector) can select one or more keys in different ways. E.g. there are simple lists, search-fields, range-selectors etc.











The selector-block



All selectors of a query are positioned inside the "Selector-area" directly under the title of the query and shown as small boxes. Selectors can be inserted, edited or deleted here. On the small symbol at the left side of a selector you can recognize its type, the text inside the selector shows its options.

Selector-types

There are several kinds of selectors, each with different possible selection sizes and different ways of displaying their options:

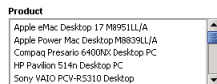
	Type	Description
	Single	Allows the selection of a single element with a drop down box.
	Multiple	Allows the selection of no, one or several elements in a list.
	Checkbox	Allows the the selection of no, one or several elements. All elements are visible at the same time and displayed as check boxes.
	Radio	Allows the selection of a single element. All elements are visible at the same time and displayed as radio buttons.
	Interval	Allows the selection of an interval with two drop down boxes, in which the user can select first and last element of the interval.
	Hierarchy	Allows the selection of a single element with a hierarchical search. The selection starts with a single drop down box, in which the user can select an element from the highest dimension-level. Then the children of the element will be displayed in a second drop down box and so on until the desired element is selected.
	Hierarchy-Tree	An interactive tree. This tree allows to navigate in the dimensions hierarchy without executing the query meanwhile.
	Button	Allows the selection of a single element with a button. The selectable element is defined by the creator of the report.
	Input	Allows the search of elements with a search pattern. The search-box accepts the wildcards "*" and "?".
	Calendar	Allows the selection of years, quarters, months, weeks or days from a classic calendar.

AB	Constant	Displays and selects a certain element and is not changeable by the user.
----	----------	---

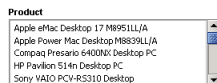
Selector options

Each selector offers a defined set of options, from which the user can select one or more elements and filters the query with. A selector does not always offer all keys of a dimension or dimension-level, only the keys defined by the options are selectable.

A Multiple-selector offering all elements of a dimensions-level



The same selector with only the three first options



A special case is the input-selector, with which the user can search keys using a search-pattern: Since this selector initially offers no keys but only an empty search-field, the options here define the restriction of the search. This means the result of a search is always limited to the keys which also occur in the options.

Alternatively you can define a boolean expression as a match-filter for a selector, which delimits the options when they are created or whenever a value is selected. Especially for search- and hierarchy-selectors this is the better way to define the options.

Default selections

Each selector can define a default selection, which is pre-selected if the user executes a query for the first time. If no default selection was defined for a selector, then the selector is empty after the first execution of the query and the selection-result of the selector is empty and that the report may show no data.

A selector without default



Selector with default



A selector of the type "single" without a standard selection contains an additional empty option as its first element, which stands for the missing selection and shows the user that this selector must be selected. After choosing an option, this empty option is omitted and disappears.

Dependence of selectors on parameters and among themselves

Usually the options and the selections of selectors do not affect each other - e.g. you can create two selectors for the same dimension and select different values with them. Also, the options and the standard selection are normally not affected by the parameters of a report.

In some cases it can be wanted that the options and the standard selection are dependant on other selectors or on parameters. E.g. imagine two selectors with which the user can select a "customer" and a "product". You could configure the second selector in that way that it only offers the products which are bought by the selected customer.

The interdependence on selectors must be activated explicitly for each selector.

Design options for selectors

Selectors offers some design options, with which you can affect their appearance:

- The **title** of a selector can be defined (otherwise the name of the offered dimension is used as title),
- the **texts** of the options can be defined (otherwise the IDs of the offered dimension-elements are used as option-texts),
- the **width** of the selector can be set (otherwise it's determined automatically on the basis of the longest option)

Automatic refresh

Single selectors and others allows the automatic re-execution of the report directly after changing their selection. Dependent on the report you can switch this feature off, in order to avoid the execution of the query between the manipulation of several selectors.

Providing date-formats for the calendar selectors

For the Calendar-Selector you must define which elements of your Time-Dimensions should be selectable and how their format is. In practice this means you have to provide the date-format of all Elements which should become selectable. There are 5 different formats you can specify. All of this formats are available as properties in the Property-Editor when selecting the selector:

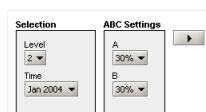
Property	Description
dayFormat	Format for days, e.g. "dd.MM.yyyy"
monthFormat	Format for months, e.g. "MMM/yyyy"
quarterFormat	Format for quarters, e.g. "q/yyyy"
weekFormat	Format for weeks, e.g. "ww yyyy"
yearFormat	Format for years, e.g. "yyyy"

The formats you provide must match the elements of your dimension, otherwise the selector won't work. If you leave out formats, the selector will not be able to select this elements.

Selector groups

Multiple selectors can be grouped together to a single selector group. Usually, selectors are placed on a horizontal row, but inside a group they are arranged vertically and therefore use less space in the report.

Selector-Groups



Like selectors, the groups allow a number of different styles which can be changed to adopt the appearance of the group. E.g. the background color, the border or the size of a group can be altered.

Outer blocks

Outer blocks (the "vertical blocks") organize the contents of a query in a vertical order. Except comments, only outer blocks may be placed directly on the worksheet of a query. Within an outer block, only inner blocks may be placed.

All elements inside an outer block (which are always inner blocks) are displayed next to each other - a vertical block arranges its contents in a row.

Adjustment (orientation) of the inner blocks

Usually the inner blocks lying in an outer block are positioned in a row next to each other. However, this behavior can be changed by the report-designer. The following adjustments are possible:

- **In a row (horizontally):** All inner blocks are displayed next to each other.
- **In a column (vertically):** All inner blocks are displayed one below the other.
- **With riders (tabbed):** The inner blocks are displayed behind each other and only one block is visible. The users can choose the block he would like to see with a tabulator.

An inner block, iterated by month and displayed as tabs

	Jan	Feb	Mrz	Apr	Mai	Jun	Jul	Aug	Sep	Okt	Nov	Dez
	Desktop		Notebook		Pocket PC		Tablet PC					
	Quantity	Amount	Quantity	Amount	Quantity	Amount	Quantity	Amount	Quantity	Amount	Quantity	Amount
Acer			2.168	5.975.775							2.059	5.156.689
Apple	4.307	11.351.102		2.044	7.706.419							
Casio							2.094	944.752				
Compaq	1.976	1.894.065	1.971	3.226.680			2.162	1.134.995			2.179	6.091.981
HP		2.016	2.328.776	2.151	5.303.952			2.026	1.504.937			
IBM				2.212	7.296.610							
Sony		2.048	5.184.116	4.025	10.480.274			1.980	1.395.771			
Toshiba				1.919	5.094.254						2.067	6.550.285

Visibility

Outer blocks can be made invisible by a condition. Invisible block will not only be invisible in the result, they also consume no computation time when the query is processed on the server.

Iteration

Outer blocks offer the possibility to repeat their contents for a set of dimension-keys (e.g. for products or months). This repetition of blocks or other elements is called "iteration".

Each block generated by an iteration is an exact copy of the original block but uses another filter (e.g. if a block repeats for a set of products, then each copy uses a filter which limits the contents of the block to a specific product).

Iteration of an outer block over months

Jan 2004			
	Quantity	Planned	Amount
Compaq Presario 6400NX Desktop PC	2.233	2.254	2.082.382
Feb 2004			
	Quantity	Planned	Amount
Compaq Presario 6400NX Desktop PC	2.023	2.063	2.019.312
Mrz 2004			
	Quantity	Planned	Amount
Compaq Presario 6400NX Desktop PC	2.104	2.153	2.019.925
Apr 2004			
	Quantity	Planned	Amount
Compaq Presario 6400NX Desktop PC	1.976	1.948	1.894.065

If an outer block contains multiple inner blocks, then all of them are copied by the iteration. E.g. a block with three inner blocks and an iteration over 10 elements produces 30 blocks in the result.

Filter

Outer blocks can additionally (like all other block elements, too) influence their filter and the filter of the elements contained in them, which affects all their expressions and properties using the filtered dimensions. The iteration mentioned above also influences the filter.

You find a detailed description of filters in the section Filters and subcubes.

Layout options

Outer blocks do not offer further layout options, since they are not visible in the report. They function as a pure grouping of inner blocks.

Inner blocks

Inner blocks may only be placed inside outer blocks. Only pivot-tables can be placed inside an inner block. The pivot-tables can be displayed in different formats, e.g. as a table, chart or newsticker, which is defined by the containing inner block.

Formatting the contents

The inner block containing a pivot table also defines the display format of its table. There is a set of different formats available for formatting the contents of a table. Tables may be formatted:

- As a simple **table**,
- as a **chart** (line-, bar- or pie-chart),
- as a **newsticker**

Also chart are simple pivot tables, which however are displayed in another format. In order to generate charts, you must know how the pivot tables are converted to them.

If a table is being displayed as a chart or newsticker, the system will convert the table into the target format using the following conversion-scheme:

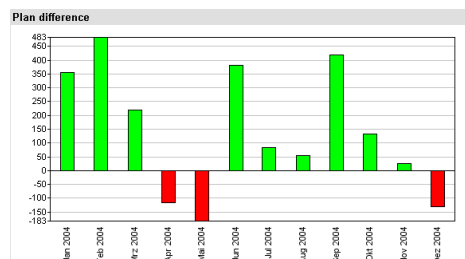
Conversion into charts

For charts, each column of the table becomes one data point, i.e. a point in a line-chart, a bar in a bar-chart or a slice in a pie-chart. Each row of the table becomes its own data series, i.e. a line in a line-chart, a bar series or a pie in a pie-chart.

Input data for a barchart

Plan difference											
Jan 2004	Feb 2004	Mrz 2004	Apr 2004	Mai 2004	Jun 2004	Jul 2004	Aug 2004	Sep 2004	Okt 2004	Nov 2004	Dez 2004
355	483	221	-117	-183	383	85	55	419	132	27	-13

Same table displayed as barchart



Links in pivot tables are automatically converted into links within the charts, so that, by clicking onto the elements of the chart, the user can jump to other reports or drilldown within a chart.

Additionally, inner blocks offer a large number of further design properties, which can be used to lay out charts and to change their appearance. With these properties almost all characteristics of a chart, e.g. the colors, the scaling, the texts etc. can be changed.

Note that the size of the inner block also determines the size of the chart. In order to display a chart, the size for the block must always be set, otherwise the chart remains invisible.

Conversion into newstickers

In a newsticker, all rows (without the headers) will be converted into a single line, starting with the first row and from the leftmost to the last column.

The input-data for a newsticker

	Jan 2004	Feb
Desktop	Desktop / Jan 2004: 20.708.942 USD	Desktop / Feb
Notebook	Notebook / Jan 2004: 46.243.658 USD	Notebook / Feb
Pocket PC	Pocket PC / Jan 2004: 4.584.120 USD	Pocket PC / Feb
Tablet PC	Tablet PC / Jan 2004: 16.179.583 USD	Tablet PC / Feb

The same table displayed as newsticker

Jan 2004: 20.708.942 USD Desktop / Feb 2004: 21.155.679 USD Desktop

Visibility

Inner blocks, as outer blocks, can be made invisible by a condition. Invisible block will not only be invisible in the result, they also consume no computation-time when the query is processed on the server.

Iteration

Inner blocks, like outer blocks, offer the possibility to repeat their contents, e.g. for a set of products or months. This repetition of blocks or other elements is called "iteration".

Each block produced by an iteration is an exact copy of the original block but uses another filter (e.g. if a block is being repeated for a set of products, then each copy uses a filter with another product and limits the product-dimension to this product).

An inner block, iterated by month and displayed as tabs

Jan 2004			
	Quantity	Planned	Amount
Compaq Presario 6400NX Desktop PC	2.233	2.254	2.082.382
Feb 2004			
	Quantity	Planned	Amount
Compaq Presario 6400NX Desktop PC	2.023	2.063	2.013.312
Mrz 2004			
	Quantity	Planned	Amount
Compaq Presario 6400NX Desktop PC	2.104	2.153	2.019.925
Apr 2004			
	Quantity	Planned	Amount
Compaq Presario 6400NX Desktop PC	1.976	1.948	1.894.065

All inner blocks generated by an iteration are lined up next to each other in a row, since they all belong to the same outer block (unless you change the the orientation of the outer block).

Filter and subcubes

Internal blocks can additionally, like all other block elements, influence their filter and the filter of all elements contained in them. All expressions and properties of the contained elements will use the changed filter. The iteration mentioned before has also influence on the filter.

Filters only affect the current selection of dimensions (i.e. the current product), but still permit to access all other, not filtered elements of the dimensions. You can also define subcubes for inner blocks, which completely limit the access to dimensions to the elements contained in the subcube.

You find a detailed description of filters and subcubes in the section "Filters and subcubes".

Linking internal blocks

Internal blocks can be provided with an additional link, which is displayed directly underneath the block as a right-justified text. Using this link you can offer e.g. further detail data for a block.

Links are displayed under the block



The link can be, like all links in HTML, displayed as a pure text or with a symbol and refer to other queries or any other URL.

Laying out inner blocks

Contrary to the outer blocks, inner blocks are visible, because they display a the frame around the pivot table or around the chart. For an inner block, you can set different design properties which affect their appearance:

- The optional **title**, which is displayed above the contents of the block,
- the **colors** of its background, frame and title,
- the **font** of the title,
- the **size** (width and height) of the block,
- the **inner padding** (the distance between the frame and the contents) of the block

A block with changed layout

Jan					
	Desktop		Notebook		Pocket
	Quantity	Amount	Quantity	Amount	Quantity
Acer			2.015	5.430.856	
Apple	3.976	10.720.472	2.188	7.803.237	
Casio					2.035
Compaq	2.233	2.082.382	1.978	3.535.250	2.019
HP	1.927	2.157.849	2.078	5.094.064	2.058
IBM			2.163	7.399.738	
Sony	2.216	5.748.238	4.135	10.548.763	2.022

The title of a block is the text being displayed directly above the pivot table. With block titles it is possible to display additional sub-titles for the individual sections of a query additionally to the main-title. Block titles are optional.

With the color-settings you can determine, how the frame around the block, the background of the block and the title should look like. If the block displays a title, then the background color of the title is equal to the frame color.

Additionally, the used font and the font-size of the title can be changed.

Pivot-tables

The actual data or a report is displayed inside "pivot tables". A query can contain more than one pivot table, but each table must be placed within its own inner block (means that you can only display one table per block).

Pivot tables are tables which determine their structure and their later contents over the definition of their row- and column-headers. Since tables own two axes (one x and one y axis), you can define such headers for both axes.

The axes of a pivot table

Product-CATEGORY	
Fact-Quantity	Fact-Amount
Manufacturer: M...	X
Manufacturer: All...	Y

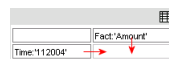
A header usually determines, which key (e.g. a product or a fact) is to be displayed. Additionally, a header can define a large quantity of properties for a whole row or column of the table, e.g. texts, formulas or colors.

Determination of the table contents

The combination of the headers determines, among other things, the contents of the table in the result and how each cell of the table is affected (e.g. the cell in the lower example is influenced the header "Fact:Amount" in the x axis and by the header "Time:'112004" in the y axis).

If a headers refers to a fact, than this fact will become automatically the fact being displayed inside the cells. All other headers determine the filter for the fact. In the lower example the turnaround for the month "112004" will be displayed, since this is the second header which has influence on the cell.

Influence of the headers on the cells (design view)



Influence of different headers on their cells

	Amount	
Okt 2004	96.594.104	
Nov 2004	87.467.274	
Dez 2004	87.123.086	

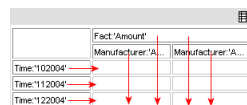
Whether facts are placed the x axis (like in the upper example) or in the y axis does not make any difference.

Encapsulated headers

The upper example used only one dimension (time) and one fact (Amount) and is very simply. It is also possible, by nesting headers, to display more than one dimension inside a pivot-table.

Nesting headers means that other headers can be placed below a header. In the lower example, two keys of the dimension "Manufacturer" were nested under the fact "Amount".

Nesting headers



When this query is executed, a table with two columns will be generated (as you can see in the upper picture, is the header containing the fact became a spanning header for the two other headers).

The cells of this query are affected, like in the first example, by their headers: The cells of the first column are affected by the header with the fact and the header with the first manufacturer and the column therefore shows the amount for the respective week and this manufacturer. The second column is also affected by the header containing the fact but by the other manufacturer. Therefore it shows the amount for the second manufacturer.

Influence of the nested headers

	Amount	
	Acer	Apple
Okt 2004	10.688.622	23.977.862
Nov 2004	10.732.414	18.273.082
Dez 2004	10.704.940	18.531.004

Nesting headers can be combined at will (at both axes) and you can theoretically display a large number of dimensions within a single pivot table.

Iteration of headers

In the upper examples each header represented exactly one key or fact. In case of a large dimension this can be very complex and uncomfortable because you had to drag single each key into the table. Also, the header would not be influenced by the selection of the user, which is

usually wanted (in the upper examples this wouldn't happen, because the headers always shows certain keys).

If several keys of a dimension are represented with individual headers, e.g. all products of a product dimension, this also has the disadvantage that new products wouldn't appear in the query before they are manually manually added to the table.

Therefore headers usually work with iterations (repetitions of keys), which are determined by formulas. Such a formula could e.g. have "all products", "the last 2 years" or other sets of keys as a result.

A table using an iteration on its y axis

	Fact:Amount
Time:MONTH	

In the upper example, the y-headers were replaced by a single header showing all filtered weeks (more exact: All filtered elements of the level "Zeit_L2" of the dimension "time"). The result the table therefore shows several rows (one per week).

The result shows all months on the y axis

	Amount
Jan 2004	87.176.503
Feb 2004	124.015.669
März 2004	86.918.576
Apr 2004	88.321.434
Mai 2004	105.090.346

In practice most headers use an iteration, even if this has only one key as result. In the first examples the simplest form of an iteration "<Dimension>:<Key>" was used which returns a single certain key of a dimension (e.g. "Time:'112004'" or "Fact:Turnaround").

If a header has no iteration, it is not repeated and results to exactly one row or column. Also, headers without iteration do not affect the filters of their cells.

Drilldown

Pivot tables offer the possibility of navigating within the headers with "drilldown". Drilldown means to "unfold" a header for the presentation of its detail data, e.g. for showing all months weeks of a year which was originally displayed in a table.

Drilldown must be activated for each individual header - headers do not offer the possibility to drill down by default. If drilldown is activated, the user will see a drilldown-symbol (a small triangle) and can unfold to the detail data by clicking onto this symbol or fold it by clicking on the symbol again.

Activated drilldown for a header

	Fact:Amount
▶ Product:CAT...	

Drilldown is also possible for all headers produced by a previous drilldown, i.e. if a user drilled down from one year to its quarters, then he can thereafter drill down from a quarter to its months and so on.

Usually, the drilldown shows the hierarchy of a dimension and all child elements of a key are visible after a drilldown (e.g. weeks for years or product for product groups). This standard behavior can be changed by the designer of a query, e.g. if he wants to display elements of another dimension when the user unfolds a header (e.g. all customers to a product).

Layout options for drilldown

There are two different layout forms for drilldown: The vertical layout and the encapsulated layout. In the vertical form all new headers are displayed below the original header after the drilldown (or next to it in the x-axis). In the encapsulated variant the new headers are displayed

nested to the original header, so that the hierarchy becomes more visible and the tables uses less rows.

Encapsulated drilldown

		Amount
▶ Desktop		524.724.708
▶ Notebook		1.36.610.138
	Acer TravelMate 2320V Notebook	187.316.768
	Apple iBook Notebook 141 M9989LL/A	83.587.564
	Compaq Presario 1215 Notebook	1.26.021.524
	HP Pavilion xt535qe Notebook PC	171.743.184
	IBM ThinkPad R40 (27232FU) Notebook	102.129.680
	Sony VAIO PCG-FRV25 Notebook	163.092.202
	Sony VAIO PCG-GRT170 Notebook	1.27.880.266
	Toshiba Satellite 1135-S125 Notebook	1.22.257.380
▶ Pocket PC		485.949.726
▶ Tablet PC		

Formulas

Till now we only created pivot-tables, in which the used fact was determined by the iteration of the headers. Instead of referring a fact in such a way you can also define a formula for a header. This is useful e.g. if you don't want to create a header for a fact (for layout reasons) or if the needed fact or computation is not available in the query-model and you need an adhoc computation.

Usage of a formula in a header

		Umsatz
▼ 1998	KW 40/1998	37.752.899,86
	KW 41/1998	37.531.905,69
	KW 42/1998	37.940.533,34
	KW 43/1998	41.312.171,86

In the upper example the formula "Amount()/Amount(YEAR)" is used to compute the proportional portion of the annual turnover. There is a large set of functions and possibilities available for the computation of values in formulas. You find a detailed description of the expression language on the "instantOLAP Reference".

Result of the formula

Jan 2004	0,078
Feb 2004	0,111
März 2004	0,079
Apr 2004	0,078
Mai 2004	0,094
Jun 2004	0,078

If you define a formula in a header, this formula will be used instead of any fact influencing a cell.

Texts and formatting

Tables offer different possibilities for the layout and format of their texts inside headers and cells.

In headers, the display-text can be set freely and has not to correspond to the key used in the header. You can also use constant texts, attribute of keys (e.g. the short-name of a product) or anything else. The text of a header can be defined with formulas, with the same possibilities like formulas.

Apart from the formulas you can also define the display-formatting for the cells, which formats the result of a formula (or fact, if you work without formulas). By formatting cells you can determine, how a number or a date is to be represented, e.g. with or without leading zeros or with a thousand-delimiter.

The upper example with title and cell-format

		% of year
Jan 2004		08 %
Feb 2004		11 %
März 2004		08 %
Apr 2004		08 %
Mai 2004		09 %
Jun 2004		08 %

Laying out pivot tables

Pivot tables offer a large variety of further layout options, both for the table itself and for the headers and cells inside the table.

A table offers general properties, i.e for the design of its upper, left corner. The headers offer properties for the layout of the headers and the cells generated by them. There are the following groups of properties:

- **Colors** of the texts, backgrounds and borders (for the corner, the headers and for the cells produced by them),
- **fonts** and **font-sizes** for the corner, headers and cells,
- **text alignment** for the corner, headers and cells,
- **sizes** (the width and height of headers),
- **rotation** of texts (vertical display of headers)
- **padding** (the inner space between the text and borders) for headers and cells

All properties of tables and headers have to be defined with formulas and everything can be computed at the time of the query execution. Therefore you can make anything dependent on certain conditions.

A table with different background, border and font-size

	% of year
Jan 2004	05 %
Feb 2004	11 %
Mrz 2004	08 %
Apr 2004	05 %
Mai 2004	09 %
Jun 2004	06 %

Comments

Comments are used to add notes to queries.

A comment below a block

Okt 2004	05 %
Nov 2004	05 %
Dez 2004	05 %

Extracted from the central datawarehouse

Like vertical blocks, comments may only be placed on the highest level of a worksheet and not within blocks or pivot tables.

Computed comments

A comment can be a static text or be computed at the time of the query execution, e.g. in order to display a comment only on a certain condition.

The computation of comments is based on formulas and you can use all expression capabilities of instantOLAP to compute comments.

Author and date of a comment

Comments can contain optional information like the author and at the time of its creation. If any of you these information is given, it will be displayed together with the comment.

Export of comments

For comments you can define if they are exported when storing snapshots (stored report results) or when results are converted to PDF or Excel documents.

If the export of a comment (e.g. a usage hint) makes no sense and may therefore be unwanted, the export for these comment should be deactivated.

Using HTML and JavaScript in comments

Comments can also be used to display HTML or execute JavaScript code in the HTML-output. By beginning the comment text with "javascript:", the comment will automatically put into a JavaScript block in the HTML-output instead of being displayed.

Filters and subcubes

Filters and subcubes are not elements of a query but very important concepts, which affect all elements of the queries and are very important for their understanding. Particularly filters have a large influence on the query-design and you should be familiar with their concept.

Filters

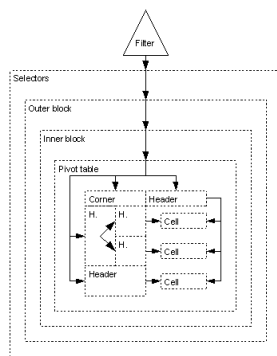
A filter can be interpreted as the "current selection" of a dimension, i.e. it determines which elements of a dimension a query, a block, a table or a header currently displays. A filter may contain as many elements of a dimension as desired.

Passing a filter within a query

Filters are passed inside a report from the top to the bottom - starting from the parameters of a query. It will be passed to the selectors, they will pass it (after they changed it) to the outer blocks, then to the inner blocks and so on. Each element of a report can manipulate or limit the filter, particularly selectors and headers in pivot tables have a large influence on the filter.

The start-filter of a reports is either a set of all root elements of all dimensions (like "All products", "All customers" etc.) or influenced by the query-parameters (if the reports was called from another query).

Passing filters inside a query



The filter of each element can be limited with the property filter of the element. Instead of using this property, the usage of the property iteration (which is used to repeat elements) is more common, particularly for headers. E.g. a header could be repeated for each week of a year, in order to produce multiple rows or columns. This iteration has a large influence on the filter, because the filter is limited for each element generated by the iteration and the manipulated filter is passed to the sub-elements of the iterated element.

A special case are the cells, which are the end of the filter chain and are always affected by two headers: By the header in the x axis and by the header in the y axis belonging to the cell. Therefore a cell represents exactly the the part of the data model which is specified by both headers (e.g. if a header with the iteration "Fact:Turnaround" lies on the a x-axis and another header with the iteration "Product:A" on the y-axis, the the cell shows exactly the turnaround for this product "A").

Influence of the filter on properties and formulas

All functions in the properties refer to the current filter the element is generated with. Particularly, if you work with relative functions inside the expressions (e.g with dimension-names or level-names), these always refer to the filter.

E.g. a header with an iteration "week" (here "week" should be a level name of the dimension "time") would return all weeks of the current time (independent if that would be one year or one day). The current time is exactly the element (or the elements) from the current filter belonging to the dimension "time".

If you use only a dimension name inside a formula, then this expression returns the elements of this dimension stored in the filter.

Note that you can access, independent of the current filter, all other elements of a dimension in formulas (e.g. if the current filter contains a "Product:A" you still can use a "Product:B" in iterations). Therefore a filter is no access limitation.

Function of the filter property

Almost any element owns a property filter, with which you can limit the filter of its sub-elements. These property accepts expressions which return dimension elements.

These dimension-elements are applied to the filter and all dimensions with at least on element in the result occur are limited to exactly these elements - all other dimension remain uninfluenced by the filter property!

Subcubes

While filters define only the current selection of the dimensions for elements and their properties, you can define real access restrictions for the dimensions with the subcube property. "Real access restrictions" means that no header inside a table has access to the dimension-elements not being part of the subcube. Therefore tables would act as if the dimension would not contain these elements at all.

Subcubes are mostly used to limit dimensions to the interesting part (usually the elements for with present data). E.g. if you want to display pivot-table where only a small part would be filled with data, a subcube is often a good way to limit the dimensions to the part with data.

rows in the chapter Pivot-tables. The functions LOOKUP and DLOOKUP (to find in the "instantOLAP Reference") also also interesting for suppressing blank rows and columns. They are used for searching dimension-elements which provide data for a certain fact and often used in subcubes.

To define a subcube you can use the property "subcube" of a pivot table. This property expects a formula which returns a set of dimension elements. All returned elements form then the new subcube (whereby more than one element for each dimension can be returned) - and all elements, which are not part of the result, do not occur in the subcube (also the facts!). You find a detailed description of the subcube property in the "instantOLAP Reference".

How instantOLAP works

Configurations

Models are not stored physically on a instantOLAP server, they are build dynamically whenever they are used the first time and then afterwards synchronized constantly.

Also, models only build their dimensions in ahead but don't need to store the values of the facts. They can be queried dynamically from the databases or they can be stored in a pre-aggregated cube - depending on the model configuration.

An administrator cannot create models directly, he can only create configurations which *describe* the structure and generation (e.g. for the dimensions) of a model. Configurations use different elements, e.g. "Key-Loaders" for the loading of dimensions or "Cubes" which describe how to load facts during the report execution.

Configurations are stored as XML-files but can be edited without the knowledge of XML using the graphical Configuration-Editor of the Workbench.

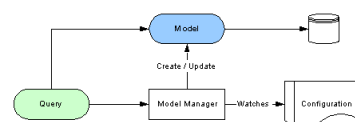
Name of a model

The name of a model always corresponds to the file name of its configuration, but without the mandatory extension ".config" of the configuration file - i.e. if a configuration named "demo.config" exists, it will create a model named "demo". A configuration "test/test.config" would create a model "test/test" and so on.

Synchronization of models

After a model has been started an created, the system will check the configuration file frequently and restart and recreate the model every time the configuration has changed.

Creation and
synchronization of
models



Also, there is also an automatic and time controlled actualization of the dimensions, caches and cubes, which is defined within the configuration. With this you can ensure that the keys of the dimensions (which represent the master data from your data sources) always keep synchronized with contents of the data sources.

Temporary models

Beside the productive models, which are created from a configuration when using a query, there are also temporary models, which are created automatically when working on a configuration in the background. You can recognize temporary models by their name, they contain the "~" sign and the name of the editing user. These models will disappear after closing the configuration editor.

Temporary models are created while you work on a new version of a configuration and test it. They will not affect the productive models, which still exist and can be used. Only after storing a configuration the productive model is replaced by the new version.

Dimensions

In instantOLAP, dimensions are build at the first startup of a model and then unless the configuration changes or the dimension is scheduled for a refresh. After a dimension was loaded, it will be used for queries without being reloaded for each query execution.

This is mainly done for performance reasons, because building a complex dimension with all its keys and attributes (which could also be assembled from multiple dimensions) uses CPU time and will not be done each time a query is executed.

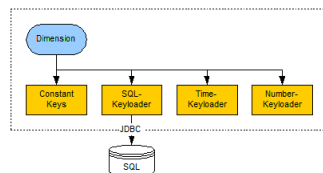
Key-Loaders

Dimensions are not simply mapped to columns of a single database - you can use and combine the loader modules for dimensions (the so-called "Key-Loaders") to build complex structures and hierarchies in your dimension and even combine data from multiple databases or other types of datasources in one single dimension.

There are several types of key-loaders which can be combined to assemble a dimension:

- **Constant Key-Loaders**, which add a single constant key to your dimension and don't need any datasource,
- **SQL Key-Loaders**, which load keys from a JDBC-datasource,
- **Time Key-Loaders**, which allow to build keys from the internal calendar of instantOLAP,
- **Expression Key-Loaders**, which allow to load keys from other dimensions into a new dimension.
- **Number Key-Loaders**, with which you can generate keys from number-ranges,

*Dimension use
Key-Loaders to collect
their keys from
different datasources*



The definition of a dimension can contain no, one or more of this Key-Loaders, both successively and nested. When loading or refreshing a dimension, they will be executed from top to bottom.

All Key-Loaders offer a large number of properties which you can use to define how keys are loaded and which attributes they will contain. Except for the very simple Number-KeyLoader all loaders share the same concepts:

The ID-expression or -pattern

The main property of a loader is the ID-expression (or pattern in the case of a Time-KeyLoader) which defines how the ID of the generated keys will look like. The ID of a key must be unique inside the complete dimension (even between different levels of a dimension).

If keys cannot be unique inside a dimension because several IDs conflict with each other they must be redesigned to become unique. E.g. if there was a product dimension with categories and products, and both levels share the same numbers as IDs, you could append a prefix 'C' to the categories and and prefix 'P' to the products.

Depending on the loader type, the expression can be a constant string, an SQL expression, a time pattern or a complex instantOLAP expression.

The parent / child relationship

A plain Key-Loader without a defined parent / child relationship would place all keys directly under the root key of a dimension. In the case you want to create a hierarchy with more than one level, you must define for each loader where to place a key by defining an additional parent expression for the loader.

The parent expression works like the ID expression but is used to search for an **existing** key inside the dimension which will then be the parent of the new key.

The parent is searched by its ID by default, but it also possible to define an attribute as the value being searched for - then the new key would be placed under the existing key owning this attribute value.

If a loader generates the same ID twice or more, this is usually no problem and the system will generate the key only once (but collect all attributes from all different loads). However, the parent / child relationship must always be the same, otherwise the system would raise an "Hierarchy violation" error, because a key must only have one parent. There is a switch to suppress this error, but usually this error indicates a problem with your dimension design and should never be suppressed.

Normally you don't have to define the parent expression for a loader manually because in the configuration editor you can nest the loaders and this will let the ID expression of a parent become the parent expression of the nested loaders. Because the nested loaders will load the children of the containing loader, nested loaders are always executed from top to bottom, otherwise the nested loaders would not find the parents for their keys.

Nesting loaders is only possible with loaders of the same type because the expressions have to be compatible. But when using manually defined parent expressions you could also place keys generated by a SQL-KeyLoader under keys generated by a Time-KeyLoader etc.

A special ability of the SQL-KeyLoader is the "recursive" mode which will re-execute the key generation until no new key has been added to the dimension. This is very useful for dimensions which are defined with a parent / child relation inside a database table, because the parent keys could be loaded from such a table after their children. Without the recursive mode these keys would be dropped.

Attributes

Each loader except the Number-KeyLoader is able to load attributes together with the keys. For each attribute defined by a loader there are some mandatory properties:

- The **name** of the attribute. The name does not need to be unique within the loader or the dimension, different loaders could load the same attribute etc.
- The **expression or pattern** which defines the attribute value. Like the ID-expression, the syntax of this expression depends on the loader type and could be a SQL expression, a time pattern or something else.
- The attribute **type**, which could be a basic type like double, integer, string etc. or a previously defined dimension (in this case the attribute would be a link to keys of another dimension).

Links can be defined as **unique**, then each value for an attribute must only exist once in the whole dimension. Unique keys are usually used as identifiers for keys (e.g. a ProductID could be an unique attribute) which are useful to map keys inside cubes. If an attribute is defined as unique, the loader will fail while loading a key with the same attribute value another existing key already owns.

The attribute values will always be casted to the attribute type. E.g. when loading a string from a database into an integer attribute, the string will automatically be casted into a number.

The loader mode

All Key-Loaders, which can add attributes to keys, have a "mode" property which exactly defines their behavior

- The mode **insert** only allows to insert new keys to a dimension. If the key already exists (it could be loaded by a previous loader or this loader could have generated the ID twice), the system raises an error and stops loading the dimension.
- The mode **extend** will only add the attributes it defines to existing keys but it will not create new keys within the dimension. This mode is very useful, e.g. to add attributes from multiple databases to keys.
- The mode **insertandextend** will both create new keys and add attribute to existing keys. This is the default mode for Key-Loaders.

Especially when working with relational databases, the extend mode has another advantage: Because the ID and all attributes are loaded with the same SQL statement, keys could be dropped if an attribute comes from another table and there is no record for these keys. You could define the joins between the tables as outer joins, but this could slow down the database and take too long.

As an alternative to outer joins, attributes can also be loaded with an additional loader which uses only the attribute table. This loader will add the existing attributes to all keys with a second statement, using no joins at all.

Time dimensions

Time dimensions are always a special case in OLAP-system. Some systems own an internal time dimension which can automatically be used in queries without having to build it before. This is different in instantOLAP and you'll have to define your own time-dimension. There is a special Key-Loader (the Time Key-Loader) which allows to access the internal calendar of instantOLAP and to generate keys from it in a very flexible way.

The reason you'll have to define your own time dimension is because you're much more flexible in the definition of your time dimension and can also use other datasources than the calendar to build it. E.g. you are able to import time dimensions from databases and only define keys for days with available data. An automatic time dimension would not be able to do this for you.

Synchronization of dimensions

Because the data inside the databases used for the dimensions can change, you can set a time pattern for your dimensions which defines how often they will be reloaded. After the reload of a dimension, the new data will be available and visible to the queries.

Reloading a dimension discards all existing keys and always re-executes all of its Key-Loaders. If one of the loaders contains bidirectional links to other dimensions, this other dimension will also be reloaded because they will be changed.

Depending on the type of data you store in your dimension, it will be necessary to reload the data more or less often. A calendar would e.g. not change more than once a day, so it would be enough to rebuild a time-dimension once every night. Other dimensions like products could change more often and should then be reloaded more frequently, e.g. once per hour.

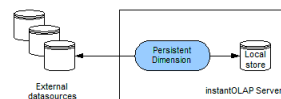
The storage of a dimension

Dimensions have a storage model. This defines, if a dimension will totally loaded, only part-wise loaded if a key is needed or not loaded at all. The persistent storage model is the default model and the best in the very most cases. The other models are only used for very large dimensions where performance issues appear.

Persistent storage

The persistent storage executes all Key-Loaders on a dimension refresh and stores all generated keys on the hard disc. After the dimension is loaded, the loaders will not be re-executed unless the next refresh of the dimension is triggered.

Dimension load their keys once and store the in memory or in a local store

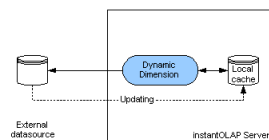


This storage model is the default model for dimensions.

Dynamic dimensions

Another storage model for dimensions is the "dynamic" storage. In this model, keys will be queried from the datasource at the first time they are used by a query and then added to an internal cache (which lies on the hard disc).

Dynamic dimension load and cache keys at the query-execution



Because this is like a "direct" mapping to a database, dynamic dimension can't be as complex as loaded dimensions. You can use only data from a single database in them and the hierarchy must be very simple.

The system always load the children of one parent together, so the hierarchy of the dimension should be rather deep than flat and no key should have more than thousand children. Otherwise the system would always load a large number of keys while executing a query, which could cause a slow and unsteady performance.

You can use dynamic dimension to import data from databases with a very large number of keys without loading it in ahead.

Adhoc dimensions

The last storage model for dimensions is the "adhoc" model. Actually, adhoc dimensions are extremely simple - they must not contain any KeyLoader and never store any key to disc, but they automatically create any key they are queried for.

Because of this automatic creation, you can map these dimensions to a database column and use them together with the LOOKUP statement. The LOOKUP will not only seek the keys with values then, it will also automatically create them on demand.

So, if you have a very large database without a hierarchy, the adhoc model could be a solution for you as they are faster than loaded or dynamic dimensions and cause no performance loss when executing a query.

Access control for dimensions

The access to keys can be limited to single users or user groups. instantOLAP offers a powerful access control, which allows to define exactly which user or group can access which keys from which dimension. Every user will only see the keys in the queries to which he has access to. All elements of queries like selectors, expression editors etc. will only display this keys.

The access control for dimensions in instantOLAP is based on rules. The administrator can define a set of rules (which are boolean expressions) for each dimension. These expressions are executed for each key a user wants to access. Only if one of the rules returns to "true" the user can view or edit the key. The rules allow both to control the viewing or editing of dimensions.

To "edit" a dimension means to edit a fact for a coordinate which contains a key of this dimension. The user can only edit the fact if he has a write access to *all* keys of the coordinate.

The expression language of instantOLAP contains a number of functions to access information about the current user executing a query. E.g. you can access the current user name or check his membership in a role.

If no access rule is defined for a dimension, all users may see and edit all keys of this dimension.

Facts

Before you can use facts in queries or create Cubes which deliver data for them, you must **define** the facts in your model.

In instantOLAP all facts are defined in a special and mandatory fact dimension. Each fact is defined as a key in this dimension, together with some additional properties like the type of the fact, its format, its unit or user-defined attributes.

Usually facts are created with Constant KeyLoaders inside the fact dimension. The workbench offers a easy way to do this. When the model is started first, the facts are loaded as keys into the fact dimension and available for the users inside their queries.

The Constant KeyLoader offers a few more properties if they are used inside a fact dimension, the "type", the "format" and the "unit". Use this properties to set the type of the fact values, the format with which they are display in queries by default and the optional unit.

Accessing facts

After a fact was defined they can be uses in queries. There are two ways to address facts and to display their values: You can place facts in table headers or filters and let the system automatically display the values. Or you can use formulas (e.g. in table headers) to define what has to be displayed in tables cells (or at any other place).

In expressions, facts can be accessed with the function CUBE, which allows to read to content of cubes with a different filter than the current. Or you can use the *Fact-Functions* generated by the model - their names are equal to the names of the facts (unless the name of a fact is not convertible to a well-defined function name!). E.g. a fact "Amount" will automatically create a function "Amount()". Using this functions is simpler and easier to read than using the CUBE function.

Where the fact data comes from

The facts defined in the fact dimension only represent the values and allow the user to place it inside queries. But when a query is executed, the system must load the values for a fact and place them in the result.

There are several sources where a value can come from:

- **Cubes** deliver and aggregate data which is available in their datasources (SQL-databases). The data can be queried in realtime or it can be preloaded and aggregated. Preloaded data is stored on the hard disc.
- **Formulas** calculate data for a coordinate (which are not available in any cube) on basis of other coordinates (e.g the Amount for a quarter could be calculated on basis of three months or a Profit could be calculated on basis of the Income and Expenses).
- **Caches** store values data are already loaded from a cube or calculated and make them available for following query-executions.

Formulas and Caches also act like Cubes though they are treated different in the configuration editor. Read the following sections to learn more about cubes.

Cubes

How data is queried from cubes

instantOLAP is a "Multicube OLAP" system and it is possible to define more than one Cube per model. Each Cube can deliver a different fact or the same fact for a different combination of dimensions or the same fact for the same dimensions but with a different filter - you can exactly define which data will be loaded from which Cube(s).

The combination of Cubes in instantOLAP allows to gather and combine data from multiple datasources. Additional Caches allow to reduce the response-time for queries.

However, outside the model cubes in instantOLAP behave as one large "Hypercube" and instantOLAP automatically decides which data has to be loaded from which cube. It is important to understand how the system determines which coordinate is loaded from which Cube and how to influence this.

Collecting the data: The ToDo list

When executing a query, at first all needed Coordinates are collected into the so-called "ToDo list". Then the complete ToDo list is processed step by step and the following operations are performed for each coordinate:

1. All existing **Caches** will be checked for the coordinate. The first matching cache will return the value if it contains it. If a cache matches is determined by their match-expression - if a cache has no match-expression it will cache and deliver all data.
2. If no cache matched, the **Cubes** will be checked for the coordinate. If a Cube matches depends on the facts and dimensions being mapped by the Cube. Also, an optional match-expression can limit the Cube. If more than one Cube can deliver the queried coordinate, the order and the match mode (see below) of the cubes will decide which cube will be used.
3. If no cube matched, the first **Formula** matching the coordinate is used. Formulas are usually defined for a certain fact and may also contain a match-expression. If a formula matches, all coordinates needed to calculate the formula are added to the ToDo list.
4. If no cache, cube or formula matched the coordinate, the fact will remain NULL.

After the ToDo list is processed and the sources for each coordinate are found, the system will load the data from the caches and cubes. Finally, the loaded values will be added to the caches.

A way to avoid the usage of ToDo lists is to use "Stores" only. Stores cubes are precomputed cubes. Because there is no need to gather all needed cells for the generation of SQL statements, the complete preprocessing including the ToDo list is skipped.

The match mode

The cubes contain a "Match mode" property which controls where the system will load the data from if more than Cube matches a coordinate. A cube matches a coordinate if:

1. The queried fact is mapped by the cube AND
2. at least one the keys of the coordinate which are no root-keys, is mapped by the cube AND
3. the match expression of the cube is empty or results to TRUE for the queried coordinate

If more than one cubes matches, the system will check them from the left to the right (beginning with the first cube defined in the configuration to the last) and check their match modes:

- If the match mode is **best**, the coordinate will also be matched against the following cubes. If any of the following cubes maps **more** dimensions than the current cube (remember they only have to map at least one dimension to match), this other cube will be used. If This is the default match mode for cubes.
- If the match mode is **incomplete**, the coordinate will be loaded from this cube but the system will **also** check the following cubes and load the coordinate from there if the current cube returns NULL for the queried coordinate. This mode is useful if you have more than one datasource and you don't know where data can be found (e.g. in a current or historical table).
- If the match mode is **first**, the first cube matching the coordinate will be used and the following cubes will not be matched against the coordinate. This setting is sometimes used to increase speed when using a large number of cubes.
- The last match mode is **exact** and will only load a coordinate if **all** keys are mapped in the cube. You can use this mode if you want to ensure that only a certain combination of dimensions must be loaded from a cube, e.g. when using preaggregated tables from a data-warehouse.

SQL cubes - feeding the SQL generator

In Version 2.5 the only possible type of cubes are SQL cubes. All other kind of data, which have previously be loadable with other types of cubes (CSV and XMLA) should now also be loaded using the SQL cubes with special JDBC drivers.

The star schema

A SQL cube is a star schema "interpretation" of the tables of a single database:

At the center imagine the "fact table" which contains the fact-values. Numerous dimension tables are linked to the fact tables, they contain usually some aggregatable columns - in difference to the fact tables, which usually only contain references to the deepest level of a dimension.

The number of possible aggregations is determined by every possible manner in which the dimensions or their levels mapped by a cube can be combined.

Mapping SQL cubes

In instantOLAP, telling the SQL cube where to find the facts and the dimensions and levels within the star schema is called "mapping".

You must map every fact you want the cube to represent and you must also map every level of every dimension you want to make these facts available for. Your goal should always be to map a complete dimension (e.g. by mapping all levels) if possible, but cubes are also capable of mapping only a part of a dimension.

The purpose of the dimension mapping is to let the SQL generator create statement which can be used to:

- assign the values of the result rows to coordinates by adding the mappings to the SELECT clause and by finding keys with IDs or attributes matching the values in the corresponding column,
- filter the result by using the mapping in the WHERE clause together with IDs or attribute values

The SQL generator can be influenced doing this in many ways, some of them are described later in this chapter. When mapping a cube there are several rules you have to consider:

1. Because a single SQL cube will try to load all mapped facts with a single SQL statement, you must **never map more than one fact table within one cube**. Otherwise the cube will generate a SQL statement which contains multiple fact tables and link them using SQL joins. If there is no real 1:1 relationship between these tables, the resulting data will be wrong (usually you get too high values then).
2. Facts are always mapped with an surrounding aggregation function like SUM, AVG, MIN or MAX. If you forget to use an aggregation, the result will only contain the first value of the SQL result instead of the aggregated value (the result would product multiple rows for the same coordinate and instantOLAP would read the first but block the following values). However using an aggregation function is not mandatory, because some value cannot be aggregated (e.g. when loading strings which will be displayed in a list view) or some databases are already preaggregated.
3. Dimensions should be mapped in tables which are as "close" as possible to the fact table in order to generate smaller SQL with less tables being used and joined together - this will result into faster statements. The optimum is to map a dimension level direct to the fact table which is usually possible with the lowest level of a dimension.

There a different possibilities to map dimensions or parts of dimensions to tables:

- You can map a complete level of a dimension to an SQL expression. This expression (which is in the simplest case a <table>.<column> expression) will then be added to the generated SQL statements and the result of this expression will be used to find keys in the dimension by comparing their IDs with the values.
- You can also map an unique attribute to an SQL expression. The system will also add this expression to the generated SQL but it will search for keys containing the result as attribute instead by their IDs. E.g. you can map an expression like "facttable.product_id" to the attribute "ProductID" of a dimension. Only keys containing this attribute will be mapped by this cube unless there is another mapping matching the other keys.
- It is also possible to combine both mapping methods and to map only keys of a certain level containing an certain attribute.
- Another and very flexible method is to map a single key to a SQL where clause. The SQL generator will then generate a single SQL statement for this key and sort the complete result to this key. This is very useful if you want to map keys to conditions - e.g. if you have "TurnaroundCluster" dimension containing a key as ">10 Mio" you could map this key to a where clause like "facttable.amount > 10000000".

All other mapping methods can also contain SQL where conditions which will automatically appended to the generated SQL whenever they are used. Mapping a single key is the only method where providing a where condition is mandatory.

The root key of a dimension usually does not need to be mapped, because the top level aggregation for a dimension in SQL is simply calculated by omitting it in the statement.

Controlling the SQL generator

The **omit** factor (a value between 0.0 and 1.0) controls whether a dimension will be filtered in the WHERE clause of a SQL statement. If the percentage of filtered keys (compared to the complete number of keys in their dimension) is larger or equal than the percentage defined by the omit factor (where 1.0 means 100%), the generator will add no WHERE clause for this dimension.

To omit a WHERE clause will return more rows for the SQL statements and instantOLAP will ignore them while reading the result. But if a WHERE clause is too big, it's often much faster to omit it and read some more data, that's why the omit factor is an important optimizing technique in instantOLAP.

The default value for the factor in a mapping is 0.75, this means the WHERE clause is normally omitted if more than 75% of the keys of a dimension are filtered.

The **operator** of a mapping lets you change the operator used in the WHERE clause. The default operator is "IN", this means the generated clause has the form "WHERE <mapping> IN (<value1>, <value2>, ..., <valueN>)" or "WHERE <mapping> = <value>" for single values.

As an alternative you can change it to "BETWEEN" which can result into less generated SQL code. If a dimension has scalar IDs or attributes (for example the number of a month in a year would be scalar), the generated clause would be "WHERE <mapping> BETWEEN <value1> AND <valueN>" where value1 is the smallest and valueN the largest value.

Like the omit factor, this will also return more result rows (which again will be ignored) but can also increase the database performance.

The match expression for cubes

Cubes can contain a boolean "match expression", which limits the values a cube will deliver. E.g. a match expression "HASKEYS(Product:A)" would let the cube only deliver data for the products A.

For cubes, the maximum complexity of match expression is limited, and only the functions HASKEYS, HASLEVEL, AND and OR are allowed. The reason for this limitation is, that the cube must be able to calculate a filter from this expression which is then applied to all result coordinates.

E.g. the example above, which only delivers data for product A, does not need any mapping for the product dimension. However, all results are for product A and the cube needs to know this and to manipulate its result.

Loading SQL data into a local "Store"

Instead of querying data from the underlying SQL databases every time a query is executed you can also turn SQL cubes into "Stores". A Store precomputes all values for all coordinates it maps by loading the data for the lowest level of the dimensions from the database and "rolling" the data up to the root key.

*If your model **only** contains Stores, there will be a significant performance boost and less memory consumption, because the system will no longer need to create and render ToDo-List.*

Before you can load a cube into Store, you must create it inside the cube and add the wanted dimensions and facts to it. For each added dimension you can define a loading mode, which accepts two different settings: "rollup" and "query":

If the mode is set to **rollup**, this mapping will be part of the initial load statement. Each dimension, which mapping has the rollup mode in this statement will then be rolled up.

The other mode **query** triggers the system to load the data but not to roll it up. You can use this mode on different levels of the same dimension or for single keys which are mapped individually. The query mode is useful for dimensions or single keys which cannot be computed by rolling up the data. The system will then perform single queries to calculate the data with database instead of doing this itself. Whenever you use the query mode, the system will automatically perform an addition query to load the root key of a dimension to calculate the root level.

Consider that if you use the "query" mode in different dimensions, the number of sql statements will increase exponential. E.g. if you load three levels of one dimension using the query mode and two levels from another dimension, this will result into 12 queries (three levels + root level * two levels + root level). The optimum is always to use "rollup" only, this will create only one statement to fill the cube.

If not every dimension is loaded into the store, it will not deliver values for the other dimensions. To avoid this, you can create more than one store for each cube and deliver data for different combinations of dimension. Or you can keep the original cube in addition to its stores, then the original cube will always deliver the data if none of its stores is holding the wanted data.

If a Store is created, its "Cron pattern" defines when and how often it will be refreshed. It will keep its current data until the cron triggers - then it reloads its data and after a successful loading the content of the Store will be replaced with the new data.

If a model is started and a Store is not loaded, the configuration property "Load Stores immediately" will force the Store to be loaded right after the model-start if set to "true". The default value is "false".

Formulas

Formulas are used to calculate facts which are not available in any cube. This can be e.g. complex calculations which cannot be performed by a database, facts which have to be computed with data from multiple databases etc.

Definition of formulas

If you want to create a new formula, there are several things you can define for it:

- The **fact** for which the formula should apply. This is optional, if the formula is not bound to any fact it will be used for all facts.
- The mandatory **expression** which will be used to compute the fact.
- An optional **match expression** which defines for which coordinates a formula should apply (e.g. you may only want a formula to calculate values for years, but not for months, days etc.).

The calculated fact

The fact is the first thing you must define when creating a new formula. Only this fact is calculated by the formula. If you want to calculate other facts you must add new formulas (for the other facts) to your model or leave the fact out.

Formula-Expressions

The expression describes how the fact should be calculated. Because instantOLAP is a multidimensional system, the expressions are also multidimensional and can apply to a large number of coordinates. In the expressions you can use other facts or navigate within the dimension, as an example you could calculate a "Year To Date" value as the sum of another fact from the past months in the current year.

The Match-Expression

The Match-Expression for a formula is optional. It defines, for which subset of the model the formula should be used to calculate a fact. If no Match-Expression is defined, the formula will apply to all coordinates of the model (for the given fact).

Match-Expression can be used to calculate different parts of the model with different formulas. Match expressions are boolean expressions. If the expression result to "true" for a coordinate to be calculated, the formula will apply. Otherwise the following formulas will be used to calculate the value (if any matches).

Match Expressions are useful to perform complex calculations. As an example, a YearToDate of the turnaround could be calculated with the following two formulas:

```
YearToDate (for the first month in a year) = Turnaround of the current
month
```

```
YearToDate = YearToDate of the previous month + Turnaround of the
current month
```

Then the first formula had the Match Expression "For the first month in a year", or written in the instantOLAP syntax:

```
HASPOSITION( MONTH, 0 ).
```

Because you can use Match Expression to calculate different subsets of the model with different expressions, there can be more than one formula for a single fact in your model.

Therefore, the order of the formulas inside your model is important. Whenever the system looks for a formula in order to calculate a value, the *first* formula matching the current coordinate is used.

Formulas can also be recursive if a formula need other values to be computed. This other value could also be calculated with other formulas or even with the same formula. E.g. the second formula in the upper example would call itself until the first formula applies because of its Match-Expression.

Caches

Caches are used to speed up the execution of queries by storing the results and re-using them for future queries as a whole or only parts of it. Caches do not store whole results but the single values for each fact and coordinate which has been used by a prior query. If a new query only needs some of the coordinates, it will then be delivered with values out of the cache. If only parts of the coordinates needed by a query are available in caches, the system will use this coordinates and query the rest from the cubes.

Caches store their values on the hard disc of the server and can become very large without using significant memory of the server. If you only use offline cubes, there is no need to use caches, because they have both the same speed.

Definition of Caches

When creating a new Cache you can define different properties for it:

- The unique **name** of the Cache
- The **maximum age** of all values stored in the Cache
- An optional **Match Expression** which defines which coordinates are stored in the Cache

The maximum age of a cache

After values are stored in Caches they can change in the Cubes which they originally come from. Therefore it is very important to define a maximum age for the entries of the cache. Any entry being older than this maximum age will be deleted from Cache (and then re-queried with the next query execution) whenever the Cache is checked for outdated entries.

How of a Cache is checked of outdated entries can also be controlled. The systems allows to define a cron pattern which exactly describes how often a Cache should be cached. Together with the maximum age, this is a very flexible way to control the contents of your Caches. E.g. you could check the Cache every 5 minutes for entries beeing older than 2 days or check it every night at 00:00 for entries older than 0 seconds (and therefore always clear it at midnight).

The Match Expression of a cache

By default a cache will store all results of all queries. But with its Match-Expression you can define for a cache, which content has to be store and which not.

This allows to set up different stores (with a different maximum age, size or cron pattern) and store different contents in them. E.g. you could set up a cache without a maximum age for facts with never will change in the datasource and another with a short maximum age for facts which change often.

The Match-Expression must be a boolean expression and can use all functions of instantOLAP to check a coordinate (which should be stored) for compliance. E.g. an expression "`NOT(EXISTS(NEXT(Period)))`" will cache all data except for the current year, month or day.

Importing other models

Model imports are used to make models easier to administrate because often used dimensions can be defined once in one or more master models and then be used in multiple other models. Therefore you need to define frequently used dimensions (like the time-dimension) only once and can use it in a large number of other models.

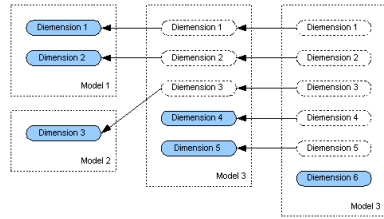
Also, using imported dimension can save system resources, because imported dimension allow to share the *physical* dimension between models. Imported dimension don't have to be synchronized by the importing models and they do not use memory or disc space in the importing model.

Imported dimensions

Each model can import an arbitrarily number of other models. From each imported model all dimensions except the fact dimension will imported and be available for Cubes and queries using the dimension.

If a imported model contains a dimension which is already defined in the importing model or a previously imported model, it will not be added to the importing model.

Imported dimensions



Importing models can be nested. If you import a model which already imports other models, your model will contain all dimensions being defined in the imported model or the models imported by this.

*Importing models will **only** import their dimensions together with their access rules. Facts, Cubes, Caches, Formulas and so on will not be transferred into the target models.*

Good model design

A good model design has an important impact on the performance of your queries and the whole system. Here are some basic rules you should have in mind when creating new models:

Make models reusable

Multiple queries can share the same model and you should really make use of this! The more configurations you create, the more complex is the maintenance of them and the bigger the system load is - because the work the server has to keep the dimension up-to-date increases with each new dimension and cube.

So every time you create a new configuration you should ask yourself if there is no existing configuration which already contains the needed dimensions and cubes. If not, you should rather consider to extend an existing configuration then to create a new one.

What should become a dimension, attribute or fact?

When turning a database schema into a model, there are many tables and columns which can be interpreted as dimensions, levels, attributes or facts. The secret of a good design is to decide, what column should become what. There are some simple rules:

1. Everything you want to make **analyzable** or **filterable** becomes a **dimension-level**. Only if there exists a dimension named "PRODUCT", you can show products on a report axis, and only with a level "COLOR" you can add a color-selector to your query and so on.
2. After you decided what to turn into levels, you should decide which levels belong together and have a 1:n relationship. If things do not belong together they should never be part of the same dimension! E.g. levels like "Product" and "Color" or "Customer" and "Order" would be different things and should not be placed in the same dimension. "ProductGroup" and "Product" on the other hand would be a good combination.
3. Additional information you may need but do not need to be selectable can become attributes **if they only depend on one dimension**. E.g. a birthdate would be a good attribute for a "Person" dimension, because the birthdate only depends on the person and nothing else. As another example, the "income" of a person would be a bad attribute, because it may change with time and therefore also depends on the "Date" dimension.
4. Information **which depends on two or more dimensions** should become facts, because only cubes can deliver data which is dependent on multiple dimensions.

Sometimes columns can become levels **and** attributes, e.g. a color could be a both without having a negative effect on your models. Often, levels which are added when extending a model to make a new query possible, exist before as attributes.

If you mix up different things in one dimension, you will often see "hierarchy violation" errors. This is a sign that there is no clear 1:n relationship between levels. In this case you should not suppress the error message and you should not try to make the IDs unique, e.g. by concatenating them (this would result into a very large dimension, because all permutations of both levels would be loaded as keys).

Naming conventions

Naming conventions can help to make models more usable and readable. Here are some conventions we use:

1. **Dimension names** start with a upper- and continue with lower-case letter. Always use the single form for dimension. Examples are: "Product", "Customer".
2. **Level names** are completely written in capital letter, e.g. "PRODUCTGROUP", "PRODUCT".
3. **Attribute names** start with a small letter, e.g. "color" or "privateAddress".
4. **Facts** start with capital letters and are valid names without empty spaces or special chars. This lets the system create valid functions for each fact. Examples are "MinAmount", "Turnaround".

This conventions are not mandatory to use and you can easily create your own. But it is recommended to invent such conventions, especially when working in a group of administrators.