Palo
Olap Server 3.0

Manual

Commercial Open Source
Business Intelligence for
Performance Management
Thank you very much for choosing Palo. Palo makes it possible to expand the functions of Excel to a great extent.

This manual of the Palo Olap Server 3.0 is composed of three parts: the installation guide, a theoretical part to introduce the concept of Palo (Palo Basics) and a hands-on section.

We recommend that you first take the time and read the theoretical part to understand the concept of Palo Olap Server 3.0. The last part describes how you actually work with Palo Olap Server 3.0.

Table of Contents

1 Introduction ................................................................................................................................. 9

2 Installing and Deinstalling Palo .................................................................................................. 10
   2.1 System requirements.................................................................................................................. 10
       2.1.1 Hardware.......................................................................................................................... 10
       2.1.2 Software.......................................................................................................................... 10
   2.2 Single Workstation PC............................................................................................................ 11
       2.2.1 Setup................................................................................................................................ 11
       2.2.2 Installation of a Local Server .............................................................................................. 24
       2.2.3 Possible Errors during Installation...................................................................................... 24
   2.3 Menu Palo ................................................................................................................................ 26
   2.4 Locations of files and databases on Windows........................................................................... 28
   2.5 The palo.ini file ...................................................................................................................... 29
       2.5.1 Client-Connection with a Palo-Server................................................................................ 29
           2.5.1.1 Palo-Server and Client on the same computer .............................................................. 30
           2.5.1.2 Palo-Server and Client on different computers ............................................................ 30
       2.5.2 Automatic backup............................................................................................................ 30
       2.5.3 Further Options................................................................................................................. 30
       2.5.4 The File palo.ini.sample................................................................................................... 32
   2.6 Connecting Palo-Clients with Palo-Server............................................................................... 40
       2.6.1 Connecting Palo-Client with local Palo-Server ................................................................ 40
       2.6.2 Connecting Palo-Client with Palo-Server in the Network............................................... 42
       2.6.3 Deleting a Server-Connection............................................................................................ 43
   2.7 Connection with the API and with the Server Browser........................................................ 44
   2.8 Backup of Databases .............................................................................................................. 48
   2.9 Deinstalling Palo.................................................................................................................... 50

3 Palo Basics.................................................................................................................................. 52
   3.1 How Palo Works....................................................................................................................... 52
   3.2 Palo is cell-related.................................................................................................................... 52
   3.3 Palo is an in-memory database............................................................................................... 52
   3.4 Palo is multi-dimensional....................................................................................................... 52

This handbook has been personalized for NA - ohneanmeldung
3.5 Working with Hierarchies
3.6 Working with Attributes

4 First Steps: Creating a Palo Worksheet

4.1 Paste View
  4.1.1 Dialog Window "Paste Elements"
    4.1.1.1 Description of the Hierarchy Buttons
    4.1.2 Selection Buttons for Element Selection
    4.1.3 Further Options in the Dialog-Window "Paste View"
  4.2 Paste Elements
  4.3 Paste Data Function
  4.4 Example of Paste Options
  4.5 Save as Snapshot
  4.6 Work Offline

5 Working with the Modeller: Developing Data Structures

5.1 The Modeller
5.2 Allowed and not allowed characters in Palo
5.3 Creating Dimensions
5.4 Creating Elements
5.5 Display of Hierarchies
5.6 Copying and Inserting Elements
5.7 Working with Attributes
  5.7.1 Creating Attributes
  5.7.2 Searching for Attributes
  5.7.3 Information Processing with Attributes
5.8 Creating New Cubes
5.9 Deleting Cubes
5.10 Additional Techniques in Working with the Modeller
  5.10.1 Comments
  5.10.2 Display of Long Comments
  5.10.3 Calculated Elements (Push Rules)
  5.10.4 Consolidation Factors

6 Entering, Changing and Deleting Data

6.1 Entering Data Manually
  6.1.1 Entering in cells
  6.1.2 Entering Data in Consolidated Cells (Splashing)
  6.1.3 Undo Data Input
  6.1.4 Copying Cell Contents (Copy & Like)
    6.1.4.1 Copy

This handbook has been personalized for NA - ohneanmeldung
6.1.4.2 Like ............................................... ................................................... ... ............................................ ................................................... ................ 234
6.2 Copying Data........................................................................................................215
6.3 Deleting Data.......................................................................................................161

7 Importing and Exporting Data.............................................................................166
7.1 Importing Data....................................................................................................166
7.2 Importing using the Data Import Wizard ............................................................166
  7.2.1 Importing Text Files.......................................................................................166
    7.2.1.1 Importing Data Cells..............................................................................166
    7.2.1.2 Update import......................................................................................175
    7.2.1.3 Importing Comments..........................................................................176
    7.2.1.4 Importing Elements............................................................................176
    7.2.1.5 Deleting Elements..............................................................................186
    7.2.1.6 Renaming Elements............................................................................188
  7.2.2 Importing from ODBC Sources.....................................................................189
    7.2.2.1 Link to ODBC Source..........................................................................190
    7.2.2.2 Importing in Palo.................................................................................190
  7.2.3 Importing from Other Cubes..........................................................................192
  7.2.4 Internal Loops................................................................................................200
    7.2.4.1 Preparatory Work................................................................................200
    7.2.4.2 Importing in Palo...............................................................................205
7.3 Additional Import Options..................................................................................206
  7.3.1 Importing from Excel-worksheets..................................................................206
  7.3.2 Importing via External ETL Tools..................................................................214
    7.3.2.1 Palo ETL Server..................................................................................214
    7.3.2.2 Cubeware............................................................................................214
    7.3.2.3 Kettle..................................................................................................214
7.4 Errors during Import.........................................................................................215
7.5 Exporting Data...................................................................................................219

8 Managing Databases .........................................................................................223
8.1 Introduction.......................................................................................................223
8.2 Managing Databases.........................................................................................224
  8.2.1 Creating New Databases............................................................................225
  8.2.2 Deleting Databases....................................................................................227

9 Application Example..........................................................................................229
9.1 Part 1: Developing of a Palo Cube......................................................................229
  9.1.1 Creating Databases....................................................................................229
  9.1.2 Selecting Databases...................................................................................230
  9.1.3 Creating Dimensions.................................................................................231
  9.1.4 Creating Elements.....................................................................................232
    9.1.4.1 Projects..............................................................................................232
    9.1.4.2 Resources............................................................................................234

This handbook has been personalized for NA - ohneanmeldung
9.1.4.3 Months ................................................................. 235
9.1.4.4 Days ................................................................. 237
9.1.4.5 Measures ............................................................. 239
9.1.5 Creating a Cube ......................................................... 239

9.2 Part 2: Creating Palo Worksheets ........................................ 242
9.2.1 Creating Project Reports ................................................. 242
9.2.1.1 Inserting Structures from Palo ..................................... 242
9.2.1.2 Inserting Charts ..................................................... 247
9.2.1.3 Entering Planned Hours ............................................ 249
9.2.2 Creating Time Entries ................................................... 251
9.2.2.1 Entering Times ....................................................... 252
9.2.3 Transferring Data Entries to Reports .................................. 255

9.3.1 Preparing of the Excel-Workbook ...................................... 256
9.3.2 Creating of the Worksheet Server Application ...................... 261
9.3.2.1 Setting up Users ................................................. 262
9.3.2.2 Data Entries by Users ............................................ 263
9.3.3 Comparing the Worksheet Server and Palo/Excel ..................... 265

10 Administration of User Rights ............................................. 268
10.1 Hierarchy of Rights ....................................................... 268
10.2 System Database ......................................................... 269
10.2.1 System Cubes and System Dimensions ................................. 269
10.2.2 Cubes in the Database “System” ..................................... 270
10.2.2.1 Cube #_ROLE_RIGHT_OBJECT ................................... 270
10.2.2.2 Cube #_GROUP_ROLE .......................................... 272
10.2.2.3 Cube #_USER_GROUP ........................................... 273
10.2.2.4 Cube #_USER_USER_PROPERTIES ............................. 273
10.2.3 Restriction of Rights in single Databases .............................. 274
10.2.3.1 Cube #_GROUP_CUBE_DATA ................................... 275
10.2.3.2 Cube #_GROUP_DIMENSION_DATA_<Name of dimension> ... 276
10.2.3.3 Cube #_CONFIGURATION ...................................... 277
10.2.4 Additional Rules ....................................................... 278
10.2.5 Applying User Rights ................................................... 280
10.2.5.1 Creating Server Entries for New Users .......................... 280
10.2.5.2 Removing Server Entries for Users ............................... 285

11 Working with Enterprise Rules ............................................. 286
11.1 General Information on Rules ............................................. 286
11.2 Creating a simple Rule .................................................... 287
11.3 Restricting the Target Area ............................................... 290
11.4 Operators of the Rule-Editor ............................................. 292
11.5 Restricting the Target Area to an Element Type ......................... 293
11.6 Drill Palo Rule ......................................................... 294

This handbook has been personalized for NA - ohneanmeldung
1 Introduction

Palo OLAP Server with Excel Add-In is a MOLAP database (Multidimensional OnLine Analytical Processing Database). It makes working with large data quantities substantially easier and faster. While a great number of worksheets are required to deal with these data quantities using Excel in its "pure" format, Palo can organize and represent data in a structured way and therefore overcome the restrictions of a two-dimensional Excel table.

Palo is an alternative to expensive software solutions as they are often used in large companies. With Palo, companies are provided a tool that allows them to easily display relevant data in a transparent manner. Furthermore, since Palo is tied into Excel, it is not very difficult to get familiar with the program.

Palo can be used in a network, i.e., the need to distribute your Excel tables is eliminated, which in turn eliminates the risk of creating various versions.

In the following, for the name Palo OLAP Server 3.0 with Excel add-in is just used the short form Palo. This product is namely the central software for analysis and planning of the Jedox AG and has given its name to the entire Palo BI Suite.

New functions in Palo 2.5 / 3.0

In version 2.5, numerous new functions and improvements were realized. The most important improvements of this version are summarized in the following overview.

- Optimized MOLAP Engine (increase of speed)
- Local data cache (speed optimization)
- Optimized execution of multidimensional cube functions
- Extended multidimensional Rule-Editor (Advanced)
- Subset Editor
- Extended subsets (named subsets, static, dynamic, private, global)
- Undo for data entries
- Zero-Suppression (suppression of zero-lines)
- Support for Drill-Through

The goal with the development of the new Palo 3.0-version was to further optimize the speed and stability of Palo. In addition, there are the following Innovations in version 3.0:

- Automatic backup with a fixed time interval or every day at a certain time
- Multithreading (only included in the Enterprise version): Processes and tasks can be distributed on multiple processors / cores
- Integration of the Palo ODBO provider in the Palo setup. It enables the creation of pivot tables with data from a Palo database
- Support for rules on attribute cubes
- Using templates for creating dimensions
- Switches for sorting dimensions and elements in the Modeller

Your assistance: Do you have any suggestions how we can further improve the program and the manual? Please do not hesitate to contact us and share your advice.

Here is wishing you lots of success in working with Palo!

Your Jedox Team
2 Installing and Deinstalling Palo

This manual describes the installation of the Palo Client as single workstation version and the local server for the Windows XP operating system.

For installation instructions for Linux, please read the readme file which is included in the Linux Server Setup download.

Please note that existing databases will not be overwritten during a new installation.

In order to reinstall the demo and also the system database you have to delete the existing ones from the data directory prior to installation. The demo database may also be downloaded directly from the Jedox website afterwards.

2.1 System requirements

2.1.1 Hardware

<table>
<thead>
<tr>
<th>Palo Client without Server</th>
<th>Minimum</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Pentium 4</td>
<td>Core Duo</td>
</tr>
<tr>
<td>RAM</td>
<td>256 MB</td>
<td>1 GB</td>
</tr>
<tr>
<td>free disk space</td>
<td>40 MB</td>
<td>60 MB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Palo Client with Server</th>
<th>Minimum</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Pentium 4</td>
<td>Core Duo</td>
</tr>
<tr>
<td>RAM</td>
<td>256 MB</td>
<td>1 GB</td>
</tr>
<tr>
<td>free disk space</td>
<td>55 MB</td>
<td>1 GB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Palo Server</th>
<th>Minimum</th>
<th>Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Core Duo</td>
<td>Core 2 Duo</td>
</tr>
<tr>
<td>RAM</td>
<td>1 GB</td>
<td>2 GB</td>
</tr>
<tr>
<td>free disk space</td>
<td>55 MB</td>
<td>1 GB</td>
</tr>
</tbody>
</table>

The recommendations may be upwards of course be exceeded. So the Palo server code can be compiled to 64 bit-architectures for e.g. Itanium processors.

2.1.2 Software

<table>
<thead>
<tr>
<th>Palo Server OS-Alternatives</th>
<th>Palo Excel Client Alternatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS Windows</td>
<td>Excel 2000 SP3</td>
</tr>
<tr>
<td>Windows 2000</td>
<td>Excel 2002 SP3</td>
</tr>
<tr>
<td>Windows XP Pro</td>
<td>Excel 2003 SP3</td>
</tr>
</tbody>
</table>

The Palo Excel client setup includes the needed software .NET Framework v2.0 and SP3 for Excel 2003 and you can optional choose this installation steps. The SP3 for the other Excel versions you can download free from the Microsoft website.

To see the Palo menu in Excel 2007, you must first install the MS Office VBA components.
2.2 Single Workstation PC

If you use an older version of Palo, please note the following instructions:

Make sure that all current changes are saved in your Palo databases by either stopping Palo-Server which ensures that all data in memory is saved or you use the Palo Service Manager and save all databases without stopping the server.

Save your data and deinstall the existing Palo version.

Note for updates to Version 3.0:
The data to be transferred must be in the format of Version 1.5. If the subdirectory that contains the previous version is displayed during the installation of Palo 3.0, all existing data will be converted into Palo 3.0 format automatically. Depending on the size of your log files and the cube structure or number of existing databases respectively, the conversion may take some time. Please do not interrupt this process.

In order for the conversion of existing database structures run as smoothly as possible, we advise that you delete all databases that are no longer needed. Please delete the *.log and *.archived-files in the respective database directory. You can find the databases in the folder Jedox\Palo\data\<database name>. Should any problems arise during the conversion, it is recorded in the PaloServer.log.

If you work with a version older than 1.5, the data must first be converted into the format of Version 1.5, i.e. after deinstallation of the previous version, you must install version Palo 1.5. After the conversion of Palo 1.5 has been completed, you can delete it again and install Palo 3.0 thereafter.

The standard setup will retain directory C:\Program Files\Jedox\Palo\data for the Palo databases. In the advanced modus the default directory for databases is C:\Documents and Settings\All Users\(Shared)\Documents\Jedox\Palo\data, but this can be changed individually.

Note: After the conversion you can find a ChangeLog.txt file under Start/All Programs/Palo. It contains a list of all new and modified functions.

2.2.1 Setup

The Palo Excel add-on of Palo 3.0 requires Microsoft .NET 2.0. If you have not already installed it, you will get the opportunity to install it during the Palo setup.

Make sure that all programs are closed, especially Microsoft Excel. Note that you must have administration rights to install Palo. Now start the Setup.exe file.

Step 1 – Select the language for the installation

If an error message appears after this 1st step instead of moving to step 2 or 10, then refer to the "Possible Errors during Installation" section in this chapter.
If Microsoft .NET 2.0 is not installed, you will be reminded. You will start the installation of Microsoft .NET 2.0 when you click YES. If Microsoft .NET 2.0 has already been installed, steps 2 to 9 are skipped and installation of Palo (step 10) is started.

Step 2 – Message if .NET 2.0 not yet installed.

Step 3 – Downloading Microsoft .NET 2.0.

Step 4 – Extracting Microsoft .NET 2.0.
Step 5 – Installation wizard of Microsoft .NET 2.0.

Step 6 – Accepting license conditions

Step 7 – Installing .NET 2.0.
Step 8 – Installing .NET 2.0.

![Microsoft .NET Framework 2.0 Setup](image1)

The items you selected are being installed.

Installation Progress:

Installing Assemblies

Assembly: C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\ILHost.dll

Step 9 – Confirmation on successful installation

![Microsoft .NET Framework 2.0 Setup](image2)

Setup Complete.

Microsoft .NET Framework 2.0 has been successfully installed.

It is highly recommended that you download and install the latest service packs and security updates for this product.

For more information, visit the following Web site:

**Product Support Center**
Step 10 – Excel Add-in and Server Setup Wizard

Step 11 – Installation Type

11.1 Standard Installation

11.2 Advanced Installation

Using advanced installation mode will enable you to reduce the installation to the local Excel Add-in for example (in that case, the server part is omitted), disable the Import Wizard and the Modeller, or add advanced control related to user management and filtering attributes.

This handbook has been personalized for NA - ohneanmeldung
Step 12 – Accepting license conditions

License Agreement
Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

This Software setup installs the Palo-Server and the Palo Excel-Add-in. The Palo-Server is licensed under an Open-Source License (GPL), the Palo Excel Add-in is freeware (not Open-Source) with a commercial license. As long as you download the software directly from http://www.jedox.com and as long as you do not redistribute the Software you can use the Palo Server and the Palo Excel Add-in free-of-charge. Our deliveries and other services are subject to our enclosed General Terms of Business for the Sale.

- [ ] I accept the agreement
- [ ] I do not accept the agreement

Step 13: Installation Options for Users

Select Installation Options
For whom should this application be installed?

Please select whether you wish to make this software available to all users or just yourself.

- [ ] Anyone who uses this computer [all users]
- [ ] Only for me [winthin]

This handbook has been personalized for NA - ohneanmeldung
Step 14 – Setting the Installation Path

If the installation folder already exists, setup will display the message on the right:

If the installation folder already exists, setup will display the message on the right:
Options of the Advanced Installation (Steps 14a – 14c):

Step 14a: Advanced Installation gives you the possibility to
- Install the Excel Add-in only
- Disable the Import Wizard and the Modeller

Please note: For Palo updates from 1.5 to 3.0, existing data in the DATA folder will be converted after the Palo-Server is being started.
Step 14c: You can change the IP address of the PC/Server or the host name as well as the port.

Step 15a (Standard installation) – You can click hyperlinks to get more info about the updates:

- [http://support.microsoft.com/kb/908002/en-us](http://support.microsoft.com/kb/908002/en-us)
or Step 15b (Advanced Installation) - You can select if you want to install the patches or not:

Step 16 – Continuation of the installation process

The Palo setup also includes an update for Office 2003. In case you are working with Excel 2003 but SP3 (Service Pack 3) is not yet installed, you should answer with yes in the dialog, with no otherwise.
Afterwards, further routines will be installed which may take a little bit longer, depending on the hardware:

Step 17: You receive a message upon successful completion of the installation.
If you have to restart your computer after the setup, a message will advise you to do so. Otherwise, the Palo settings window appears after the installation. Here you can also determine whether Palo should start automatically with Microsoft Excel.

You can find “Palo” via the start menu after standard or advanced installations. You can change settings under “Palo Settings” at any time.

The link to Palo Excel Add-in should now be on the desktop.

The task bar displays the Palo Service Manager icon and the status of the Palo service:

- Palo has started
- Palo was stopped

If you right-click the icon, you will see the settings dialog of the Palo Server Service.
Palo Server Service Manager enables you to start and stop the Palo service and to save database changes. The Service Manager starts normally automatically. Automatic start can be deactivated.

You can set the check interval of the service status under Options. It indicates how often the status of Palo-Server should be checked (values from 3 to 60 seconds are possible).

Under Start/All Programs/Palo/Palo Settings you can determine whether the Palo Add-in should always start together with Excel. Here you also select the language for the Add-in. You can make these settings at installation. If you want to change the settings later, open the Palo settings via Start/All Programs/Palo/Palo Settings.

If you defined that Palo should be started manually, start Palo by clicking the desktop icon or open it via Start/All Programs/Palo/Palo Excel Add-in.

After you have completed your settings, Microsoft Excel must be restarted. Then you will find in the Excel Palo Pulldown-menu.
2.2.2 Installation of a Local Server

A local server is installed on your computer when you install the Palo client. You can log into the local servers of other computers in a network. In practice, this makes each local server public. You can create users and groups and also assign passwords to manage necessary access rights to your server. But you can't add another server on the same computer.

After the installation, the server is configured completely. The API interface is available via the browser http://localhost:port [default: http://127.0.0.1:7777].

The access from other computers on a Palo network server is described in the chapter "Connecting Palo-Clients with Palo-Server".

2.2.3 Possible Errors during Installation

Installation will fail, if the user does not have the required user rights. You should be registered as administrator when you install Palo.

Please note that if you just deinstalled a previous version of Palo, it can be that you must restart your computer to complete the deinstallation. If the deinstallation isn't completed, the following error message appears during installation:

If you try to install Palo without deinstalling a previous version first, the following error message appears:
If you try to install Palo while Microsoft Excel is running, the installation aborts with the following error message:

![Setup](image)

Please deinstall the existing Palo version or close Microsoft Excel, and then repeat the installation. If you have not installed .NET Framework 2.0, Palo will remind you. If you click NO at this point, installation will abort.

![Setup](image)

If you would like more information, please use our forum [http://www.jedox.com/community/palo-forum/](http://www.jedox.com/community/palo-forum/).
2.3 Menu Palo

After a successful Palo installation you will find the menu Palo in the menu bar:

![Menu Bar with Palo Menu Items]

Here is the description of the menu items:

- **Paste View:**
  The dialog "Paste View" assists in generating a Palo database view with a few mouse clicks only.

- **Paste Elements:**
  Instead of the Paste View function, you can insert individual elements into your worksheet if you use the Paste Elements function.

- **Paste Subset:**
  This assistant supports you in generating and inserting subsets.

- **Paste Data Function:**
  Use this function to insert proper formulas for data retrieval from a Palo database.

- **Modeller:**
  Use the Modeller to structure the data and to create cubes, dimensions, elements, consolidated elements and attributes. Here you also find the button "Start Palo Wizard".

- **Data Import:**
  Use this function to import data from text files, OBDC databases or other cubes, for example.

- **Save As Snapshot:**
  The data (values) of a worksheet can be saved in an offline Excel file.

- **Work Offline:**
  Work without a connection to the Palo-Server.

- **WikiPalo.com:**
  Open the WikiPalo.com home page.

- **About:**
  Displays information about the developers of Palo.

Many of these menu items are also available in a Palo toolbar:

![Palo Toolbar]

Additionally, with Palo version 2.5 / 3.0, new functions are in the Palo toolbar. These will be explained to you in chapter 6 "Entering, changing and deleting data".

This handbook has been personalized for NA - ohneanmeldung
Also the context menu, which you get with right-click on an Excel cell, received extra items by the Palo installation:

- **Paste View:** The dialog “Paste View” assists in generating a Palo database view with a few mouse clicks only.

- **Drill Palo Rule:** To get information on the calculation instruction, if such a calculation statement (Rule) in the displayed Palo-cell exists.

- **Drill Through:** This item isn't further described in this manual because for its functionality the supervision-server of Jedox is needed and its operation will be explained by our consultants if this is required.

- **Drill History:** This item is reserved for future developments of Palo and it will still not work in Palo 3.0.

- **Palo Error:** If a Palo function displays an error in a cell, then you can use this menu item to get additional error messages.
2.4 Locations of files and databases on Windows

The location of the databases (=location of the palo.ini) is set during the installation process. The default folder of the "Standard-Installation" is

C:\Program Files\Jedox\Palo\Data

and the default folder of the "Advanced Installation" is

C:\Documents and Settings\All Users\Documents\Jedox\Palo\Data.

After completion of installation you will find the info of the location under Start - Control Panel - Administrative Tools - Services the service Palo Server Service. When you double-click Palo Server Service you see in the Palo Server Service Properties dialog box the "Path to executable":

At the end of this line you will find the path where the databases and palo.ini are stored:

Path to executable:
"C:\WK1\palo.exe" --start-service --service-name "PALOServerService" --data "C:\WK2"
With regedit, you will find more information, such as where PaloServer.log is stored:
If you Palo installed for all users, under HKEY_USERS\DEFAULT\Software, if you installed Palo only for
the current user, then under HKEY_CURRENT_USER\Software, and then in both cases below
Jedox\Palo\XlAddin:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Default)</td>
<td>REG_SZ</td>
<td>(value not set)</td>
</tr>
<tr>
<td>Active</td>
<td>REG_DWORD</td>
<td>0x00000001 (1)</td>
</tr>
<tr>
<td>ActiveCube</td>
<td>REG_SZ</td>
<td>Sales</td>
</tr>
<tr>
<td>ActiveDB</td>
<td>REG_SZ</td>
<td>Demo</td>
</tr>
<tr>
<td>ActiveItem</td>
<td>REG_DWORD</td>
<td>0x00000001 (1)</td>
</tr>
<tr>
<td>ActiveLanguage</td>
<td>REG_SZ</td>
<td>en-US</td>
</tr>
<tr>
<td>ActiveServerID</td>
<td>REG_DWORD</td>
<td>0x000000001 (1)</td>
</tr>
<tr>
<td>AdvancedControl</td>
<td>REG_DWORD</td>
<td>0x000000001 (1)</td>
</tr>
<tr>
<td>ConfigFilePath</td>
<td>REG_SZ</td>
<td>C:\Documents and Settings\All Users\Documents\Jedox\Palo</td>
</tr>
<tr>
<td>DisableModeller</td>
<td>REG_SZ</td>
<td>99a094bcf381e65754032830994ec86</td>
</tr>
<tr>
<td>ExcelPath</td>
<td>REG_SZ</td>
<td>EXCEL.EXE</td>
</tr>
<tr>
<td>LogFilePath</td>
<td>REG_SZ</td>
<td>C:\W1\Log</td>
</tr>
<tr>
<td>TemplatePath</td>
<td>REG_SZ</td>
<td>C:\Documents and Settings\All Users\Documents\Jedox\Palo</td>
</tr>
</tbody>
</table>

2.5 The palo.ini file

The configuration of Palo can be changed in the file palo.ini, which is located in C:\Program
Files\Jedox\Palo\data by default. This directory also contains a sample file, palo.ini.sample, which
explains the available parameters briefly.

2.5.1 Client-Connection with a Palo-Server

In order to access a Palo-Server over the network, you have to modify the file palo.ini in the directory
...\data of the Palo-Server (see chapter "Locations of files and databases on Windows").

In the palo.ini file you'll find the line

http "127.0.0.1" 7777

These are the default settings. They mean that the http-interface of the server (over which client
connections are made) expects connections over IP-address 127.0.0.1 (the standard IP-address of your
computer, also called "localhost") on port 7777. These settings can be altered already during the
installation of Palo. Bear in mind that this IP address does not specify the address(es) of the client-
computer(s). Rather it is established here over which IP-address(es) access to the server is made
possible. Therefore the meaning of the entry 127.0.0.1 (or "localhost") is, that access to Palo is only
possible from the server itself.

If you enter the network-IP of the server (e.g. 192.168.2.100) instead, other computers will have access
to Palo over this IP, but access of the server itself over the localhost connection would not work any
more – the network-IP would also have to be used in the Palo-Client.

If only two empty quotation marks are declared instead of an IP-address, then access over all IP-
addresses of the server are permitted (e.g., over different network cards and also over "localhost").
Also several connections can be declared one after the other, for example

http "127.0.0.1" 7777
http "192.168.2.100" 7778

However, different connections must be declared with different port numbers. The port number can be
selected freely but it cannot yet be in use and must moreover possibly be released in the firewall.
Quotation marks must be used only if a server name contains blanks or special characters or in case
that the server name should be left out completely.

This handbook has been personalized for NA - ohneanmeldung
2.5.1.1  **Palo-Server and Client on the same computer**

If the access should take place only from an Excel-Client, which is installed on the same computer as the Palo-Server, it is sufficient to enter the following line in palo.ini

```plaintext
http "127.0.0.1" 7777
```

This setting is done automatically during the installation of Palo, provided it is not altered explicitly in the corresponding dialog.

2.5.1.2  **Palo-Server and Client on different computers**

Suppose the computer on which the Palo-Server is installed has the network IP-address 192.168.2.100 and associated name palosv0001.

In order to make communication with the Palo-Server possible for other computers, open the palo.ini file of the server with an editor. Then perform the following modification:

```plaintext
Below the line:  

http "127.0.0.1" 7777

Insert the following new line:  

http "192.168.2.100" 7778
```

Save the file and then start the Palo service of the server again. Later, you can register a new server with these connection data in the Palo-Client. If the connection doesn't work, then you please unlock the corresponding port in the firewall.

2.5.2  **Automatic backup**

In the palo.ini file you can specify that the data from the Palo databases which are in memory of the server are automatically backed up to the hard drive.

Therefore you have two options:

**Automatic backup with a fixed time interval:**

Entry generally palo.ini: autosave T HH:MM  
Example entry in palo.ini: autosave: T 04:00

**Automatic backup every day at a certain time:**

Entry generally palo.ini: autosave L HH:MM  
Example entry in palo.ini: autosave: L 02:30

2.5.3  **Further Options**

The "palo.ini" file manages all options which are communicated to PALO at start-up. When you start PALO as server service, this file is used in any case. When you start Palo from the control panel, you can add the option "-n", so that only the options in the command line are used and not "palo.ini". If any of the options are not listed, either in the command line or in the "palo.ini", a default value is used.

We distinguish between three types of options. The first option type refers to "switch parameters". If any of the options is given more than once, e.g., one time in the command line and one or two times in the "palo.ini", the value is switched to "on" or "off" each time. The following parameters are examples of "switch parameters":

- Chdir
  - Auto-load
The second type is the options with additional parameters. In this case, the last value read by PALO is used. Be advised that the command line options and then the "palo.ini" are read if both are used. The options of the third and last group are an exception to that. Here are two examples of the options listed above:

- worker
- workerlogin

The third and last type refers to the options which collect additional parameters and use all of them. This means that in case the option is entered several times with parameters, each parameter is used. For example, these include:

- admin
- http

Command Line Parameters
These parameters can be abbreviated using individual letters. If you start Palo on the command line with "-?", these abbreviations are listed. The complete names should be used in the configuration file.

Example for Using Options
1. "add-new-database" is a switch parameter. If we enter this parameter in the command line and in the "palo.ini", it is switched on, because it is the default value for this option and it is switched off upon first entry in the command line and then switched on again in the configuration file.
2. "workerlogin" is an option with additional parameters. For example, if we enter the same one twice in the "palo.ini", only one option, the last one in the file, is used with its parameters.
3. The option "http" is additive. In other words, if you enter "http" including parameters several times, e.g. once in the command line and once in the configuration file, the parameters of both entries are used.

Switch Options:
- "auto-load": Determines whether Palo should automatically load all in file "palo.csv" listed databases that are contained in the directory, set by "data-directory". Default: true
- "auto-commit": Determines whether all changes made to databases in memory should be saved by Palo to the hard disk when shutting down the Server. Default: true
- "chdir": Determines whether Palo should switch to the directory, which was set by "data-directory". All paths listed in the configuration file are relative to the directory in which Palo is executed. Consequently, this is either relative to the directory where the binary file is stored, or relative to the "data-directory". Default: true
- "add-new-databases": Determines whether Palo automatically recognizes directories containing OLAP databases in the "data-directory", loads these databases and adds them in palo.csv.
- "user-login": Determines whether users must authenticate themselves to log on, i.e., enter a user name and password. Default: false
- "use-cube-worker": Determines whether a "cube worker" is used. Default: false
- "load-init-file": Only in the command line. Determines whether a configuration file is used. Default: true
Installing and Deinstalling Palo

Options with Non-Additive Parameters:

- "data-directory": Path to data directory that contains the databases and "palo.ini". Default: "Data"
- "template-directory": Path to directory that contains online documentation. Default: "Api"
- "log": The file that Palo log is written to. If you enter "+" as the parameter, log writes to stderr, if you enter "-", to stdout. Default: "-"
- "verbose": Determines how much information Palo writes into the log file.
- "worker": Complete path to worker binary file. This contains the name of the executable file and possible worker arguments.
- "workerlogin": Determines how much of login tasks are handled by worker. Default: No "worker-login" at all: Palo handles all login tasks.
- "init-file": Complete path (incl. name) to the configuration file.

Options with Additive Parameters

- "admin": Determines the ports to server browser and online documentation (Important: The admin interface does not require any authentication and thus should be used only in safe networking!)
- "http": Determines the ports that allow general access to Palo
- "legacy": Determines the ports which allow you to access with the older legacy interface.

The default setting of the current ini file is as follows:

Cache-barrier 5000

user-login

Verbose info

log "C:\Programs\Jedox\Palo\Log\PaloServer.log" //Path of the Server.log file
http "127.0.0.1" 7777
template-directory "C:\Programs\Jedox\Palo\Api"

If you want to use the Palo 1.0 Excel Client you must use the Legacy interface to connect the new 2.0 server on port 1234 (default) or 1230.

## legacy port for Palo 1.0 clients
#legacy    " " 1230
legacy    "127.0.0.1" 1234

2.5.4 The File palo.ini.sample

###
### PALO configuration file
###

# Configuration file info:
# - the command line arguments are evaluated first, the configuration file is evaluated after the command line arguments have been processed
# - if you start palo with -n or --load-init-file on the command line, then
#   the init file is not read. The load-init-file options is ignored if given

This handbook has been personalized for NA - ohneanmeldung
- a comment starts with a "#" sign

- parameters without additional parameters like "auto-load", "auto-commit"
  or "chdir" toggle a state from "true" to "false" and vice versa. You can
  declare a "toggle" parameter more than once.

- if parameters with additional parameters like "worker" or "workerlogin"
  are given more than once on the command line or the configuration file
  only the last definition is valid.

- the parameters "admin", "http" are treated specially. All the
  definitions supplied on the command line / in the init file are used.

- For example:

  The default of "add-new-database" is true (see palo -?). If you supply
  --add-new-database on the command line but not in the configuration file,
  then the option will be set to false. If you supply add-new-database in the
  configuration file but not on the command line then the option will also be
  set to false. If you supply --add-new-database on the command line and in
  the configuration file, then the option will be true again, as it is
  toggled twice.

  The option "workerlogin" has one additional argument. If you supply
  workerlogin on the command line and on the configuration file, then the
  value supplied in the configuration will be taken.

  If the http option is supplied for port A on the command line and for port B
in the configuration file, then both ports A and B are used.

### General options

### path of the data directory
( default directory is "Data")

data-directory c:\olap\palo-2.0\Data
data-directory /home/palo/palo-2.0/Data

### directory of online documentation
( default directory is "Api")

template-directory Binary/Api

### load all databases on server start into memory which are defined in the
palo.csv (default state is "true")

### commit all changes on server shutdown
( default state is "true")
# auto-commit

## change into data directory
# if true, palo changes into the data directory at the program start. Any path
# supplied in the init file is therefore local to the data directory in this
# case.
# (default state is "true")
# chdir

## try to add directories with OLAP database automatically and add them to
# palo.csv (default state is "true")
# add-new-databases

## force users to login
# (default state is "false")
# user-login

## splashing limits in mega byte
# generate an error if splashing requires more space than the first number
# generate a warning entry if splashing requires more space than the second
# number generate a info entry if splashing requires more space than the third
# number
# splash-limit 1000 500 100

## Goalseek algorithm can be executed on slices with maximum <cell_limit> cells (default 1000).
# Algorithm must complete within <timeout> milliseconds (default 10000).
# goalseek-limit <cell_limit>
# goalseek-timeout <timeout>
# goalseek-limit 1000
# goalseek-timeout 10000

## maximal growth in area computation

### Server address and port for http interface
### an address can be "", a server name or an internet address
### example address
### "" (for all server internet addresses)
### localhost
### server.domain.net
### 192.168.1.2
### a port is a number
### http interface with server browser and online documentation

This handbook has been personalized for NA - ohneanmeldung
# admin "" 7777
# admin localhost 7777
# admin server.domain.net 7777
# admin 192.168.1.2 7777
# admin localhost 7770
# admin localhost 7777
#

## http interface
#
# http "" 7779
# http localhost 7779
#

## keep trying
# If the HTTP port cannot be opened, keep trying until it can be opened. Default
# is to give up and exit, if the ports cannot be opened.
#
## keep-trying

### Logging
### ############################################### #############################

## Logfile
# log to stdout (default)
# log -
# log to stderr
# log +
# log to a file
# log /var/log/palo.log
#

## Log-level
# levels
# error (default)
# warning
# info
# debug
# trace
#
## verbose error
#
# verbose info

### Worker
### ############################################### #############################

## path to the worker
# worker <executable> <argument_1> ... <argument_n>
#

This handbook has been personalized for NA - ohneanmeldung
# worker          /usr/bin/php5 /home/palo/worker.php
#

## use a worker for login
# if not set, then no login worker is used
# possible values:
#    information
#    authentication
#    authorization
#
# workerlogin   authorization
#

## use workers
# (default state is "false")
#
# use-cube-worker
#

## enable cell drillthrough
# (default state is "false")
#
# enable-drillthrough
#

### Cache

## The server side cache is limited to the <total_cache_size_in_bytes>. An aggregation is cached only if the corresponding sub-cube contains more than <number_of_base_elements_in_aggregation>. After the <number_of_cache_invalidations>, the cache will be cleared completly. A cache invalidation is the change or deletion of a cell value, regardless of its type (base or aggregation). Note that setting a aggregated cell will change many base cells. If the number of changed base cell is larger than <number_of_cells> the cache will be cleared completly.
#
# cache-barrier <number_of_base_elements_in_aggregation> (default 5000)
# cache-size <total_cache_size_in_bytes> (default 100000000)
# clear-cache <number_of_cache_invalidations> (default 5)
# clear-cache-cells <number_of_cells> (default 1000)
#
# In order to deactivate the cache completly set "cache-size" to "0". This will not deactivate the rule cache.
#
# cache-size 0
#

## The size of serve side cache for rule results in bytes.
#
# rule-cache-size <total_cache_size_in_bytes> (default 100000000)
#
# In order to deactivate the rule cache completly set "rule-cache-size" to "0". This will not deactivate the aggregation cache.
#
This handbook has been personalized for NA - ohneanmeldung
# rule-cache-size 0
#

### Undo
### # In a locked cube area it is possible to undo changes. Each lock can use
### <undo_memory_size_in_bytes_per_lock> bytes in memory and
### <undo_file_size_in_bytes_per_lock> bytes in files for storing changes:
### #
### undo-memory-size <undo_memory_size_in_bytes_per_lock>
### undo-file-size <undo_file_size_in_bytes_per_lock>
### #
### undo-memory-size 10000000
### undo-file-size 100000000
### 

### SSL
### # Set the encryption type
### # none
### # optional
### # required
### #
### # If none is selected, then no HTTPS support is available. If optional is
### # selected, then you can use HTTPS. If required is selected, then only
### # /server/info will function unencrypted. All other functions require a HTTPS
### # connection.
### #
### # If you specify optional or required each http or admin command must be followed
### # by a https command containing the HTTPS port to use.
### #
### # encryption none
### 
### # Set the https port
### #
### # If you specify optional or required as encryption type, then each http or
### # admin command must be followed by a https command containing the HTTPS port to
### # use.
### #
### # https 7778
### 
### # The following certificates are need: a root certificate and a private
### # certificate signed be root authority. In addition a file with the
### # diffie-hellman parameters must be supplied. The private certificate can be
### # protected by a password. The paths are relative to the data directory.
### #
### # key-files <ca> <private> <diffie-hellman>

This handbook has been personalized for NA - ohneanmeldung
# password <private-password>
#
# The files can be generated using the OpenSSL command line tool, see the
# OpenSSL documentation for details. Generate a self-signed certificate use:
#
# Root-Certificate and Server-Key (<ca> and <private> are identical):
# openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout server.pem -out server.pem
#
# Diffie-Hellman parameter file:
# openssl dhparam -2 -outform PEM -out dh1024.pem 1024
#
# key-files server.pem server.pem dh1024.pem

## location of directory containing the HTTPS extension
#
## extensions <directory>
#

### FOR DEBUGGING ONLY, DO NOT USE IN A PRODUCTION SYSTEM
###
### generates a default session id 0000
#
## fake-session
#

### Autosave
###
## autosaves all the databases
#
## autosaves at an exact time once a day.
## autosave T HH:MM
#
## autosave T 23:59
#
## autosaves cyclically when a given amount of time elapses.
## autosave L HH:MM
#
## autosave L 10:10

### Threading
###
## initial-thread-pool <initial-pool-size> sets how many concurrent users the
## server is prepared to serve immediately after startup. after reaching this
## limit the server will dynamically increase the pool size
#
## initial-thread-pool <initial-pool-size>
## initial-thread-pool 4
#
## max-subjobs <max-subjobs-size> sets the number or processor cores allocated
## for calculating complex jobs
#
## max-subjobs <max-subjobs-size>
## max-subjobs 2

This handbook has been personalized for NA - ohneanmeldung
Q: How do these options relate to one another? Is a subjob a "child" of a thread so that 4 threads + 2 max-subjobs = 8 subjobs?
A: Yes. However, this applies only to computational intensive jobs. More precisely to retrieving aggregated cells and only when rules are involved. In such a case, a job (the query that requested aggregated cells) will be split into multiple subjobs.

Q: Is the number of possible (parallel) subjobs limited by the number of cores?
A: One can split this into as many subjobs as desired. However, as a first hint, going for more than the number of processors (times two if the processors support hyper threading) might not lead to performance improvement.

Q: Is there still a differentiation between processors and cores?
A: Not. From the software standpoint there makes no sense to do so and it was removed.

Q: Is there a limit for the threading pool after which the server won't further increase it?
A: Not. The server will always increase the pool to accept new clients. When no available threads remain to serve an incoming connection, the server will add the exact number of threads specified in the initial thread pool. On the other hand, if 50% of the threads in the pool do not serve active connections and the number of threads in the pool is larger than the initial size (the pool was increased at least once), the server will remove threads till it reaches one of the two criteria specified above (less that 50% threads idle or initial size reached).

Q: What's the advantage of having more initial threads if the server will anyway increase it dynamically?
A: The best operating situation is when the server does not adjust the thread count and all of them are busy. More threads means more memory consumed and some overhead. Therefore it is best to have as little as possible. However, creating a new thread means time spent by the server on something else than serving a client connection so it is good to have one already created. There is always a balance to be reached between memory overhead and responsiveness. On a large enterprise environment it might be a good idea to start with plenty of threads, on a simple, desktop only installation one might be enough.

Q: "More precisely to retrieving aggregated cells and only when rules are involved." Does this mean: when those two conditions combined are true, or when one of them is true?
A: Yes, both of them have to be met.

Q: What are the default settings for these options, 4 and 2?
A: initial pool size = 10, maximum subjobs = 2
2.6 Connecting Palo-Clients with Palo-Server

By default with a client + server installation is only registered the connection to the server for the user, who is logged in during the installation. Other users first must register once the connection to the Palo-Server for using it. Then it is in the server list until unregistered again.

2.6.1 Connecting Palo-Client with local Palo-Server

In order to register a new server under "localhost", you click on "Modeller…" in the Palo menu: In the following dialog window you click on the button "Start Palo Wizard …" (red arrow):

You get the dialog window "Palo Wizard":

Here you click on "Register new server" and then on "Next".

This handbook has been personalized for NA - ohneanmeldung
Installing and Deinstalling Palo

In the next step, you enter a name for the connection and enter the remaining data. Make sure that you use a created user’s name only. By default, there exists the user “editor” with the password “editor”. We enter the data as shown below:

Click on “Test connection”.

You get the following message, if the test was successful:

Confirm the message with "OK", then click on "Finish" please. You get the confirmation that the server was successfully registered.

A click on "OK" closes the Palo Wizard.
2.6.2 Connecting Palo-Client with Palo-Server in the Network

Suppose the computer on which the Palo-Server is installed has the network IP-address 192.168.2.100 and associated name palosv0001.

In order to make communication with the Palo-Server possible for other computers, we need in the palo.ini of the server

<table>
<thead>
<tr>
<th>below the line:</th>
<th>http &quot;127.0.0.1&quot; 7777</th>
</tr>
</thead>
<tbody>
<tr>
<td>the following line:</td>
<td>http &quot;192.168.2.100&quot; 7778</td>
</tr>
</tbody>
</table>

Then we can establish a connection with the Palo Wizard of the Palo-Client using the following entries:

As you can see, the name in the field "Connection name" is not required to be the network name of the server (palosv00001). Of course, the user "Smith" and his password must have been created before in the Palo-Server. Details about the creation of users in Palo can be found in the chapter "Administration of User Rights" in this manual.

Since the original entry - http "127.0.0.1" 7777 – has not been replaced, the server machine has still access through the localhost connection.

Remark: The connection is established faster, if you use the IP-address of the server (192.168.2.100) instead of the name (palosv0001) in both the palo.ini file and in the Palo Wizard.
### 2.6.3 Deleting a Server-Connection

In the "Palo Wizard" dialog window you also can delete (unregister) a server-connection:

Confirm your selection by clicking "Finish"

Then the system asks you to confirm the deletion by clicking "OK".

Palo then reports: "Server successfully unregistered".

For each server entry, created with the Palo Wizard, there exists an entry in the registry. This is explained in detail in the chapter "Administration of Palo".
2.7 Connection with the API and with the Server Browser

Beside the connections from Excel-Clients, it is also possible to define connections to the Admin-Interface of the Palo-Server in the palo.ini file. This is addressed with the prefix "admin" instead of the prefix "http".

For example, for a connection from the server machine itself:

```
admin "127.0.0.1" 7778
```

The port number must be different from other port numbers used!

**Important:** The admin interface does not require any authentication and thus should be used only in safe networking!

Over the Admin-Interface and the http-API of Palo, the server can be controlled and data can be retrieved from it, from other computers as well as from the server itself. This is even possible without an Excel-Client installed. For this purpose you have to enter the server IP address, followed by the port number, in your browser. For example, for access from the server itself: "http://127.0.0.1:7778/" or "http://localhost:7778/". There you find the start page of the API:

![Palo Olap Server](image)

**PALO server 3.0.0**

This is the open source OLAP database PALO 3.0.0.

**HTTP API**

The PALO server provides a HTTP API for getting and setting data. For more information about the API click here.

**Server Browser**

You can find the server browser here.
You can use the server browser for the administration of the databases via a browser. For example, you can delete a database via the browser.
If you click the database, you get an overview of the cubes and dimensions. Sample screens are shown below. Your content may look different.

### PALO server browser

**Database**

<table>
<thead>
<tr>
<th>ID</th>
<th>name</th>
<th>status</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Demo</td>
<td>loaded/saved normal</td>
<td></td>
</tr>
</tbody>
</table>

**Dimensions**

<table>
<thead>
<tr>
<th>ID</th>
<th>name</th>
<th>num elements</th>
<th>max level</th>
<th>max indent</th>
<th>max depth</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>#<em>GROUP</em></td>
<td>10</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>system</td>
</tr>
<tr>
<td>1</td>
<td>#<em>CUBE</em></td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>system</td>
</tr>
<tr>
<td>2</td>
<td>#<em>Month</em></td>
<td>17</td>
<td>2</td>
<td>3</td>
<td>2</td>
<td>normal</td>
</tr>
<tr>
<td>1</td>
<td>#<em>Years</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>attributes</td>
</tr>
<tr>
<td>1</td>
<td>#<em>Years</em></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>attributes</td>
</tr>
</tbody>
</table>

**Cubes**

<table>
<thead>
<tr>
<th>ID</th>
<th>name</th>
<th>dimensions</th>
<th>status</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>#<em>GROUP_CUBE</em></td>
<td>0,1</td>
<td>loaded/saved</td>
<td>system</td>
</tr>
<tr>
<td>1</td>
<td>#<em>Month</em></td>
<td>3,2</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
<tr>
<td>2</td>
<td>#_GROUP_DIMENSION_DATA_MONTHS</td>
<td>0,2</td>
<td>loaded/saved</td>
<td>system</td>
</tr>
<tr>
<td>1</td>
<td>#<em>Years</em></td>
<td>5,4</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
<tr>
<td>4</td>
<td>#_GROUP_DIMENSION_DATA_YEARS</td>
<td>0,4</td>
<td>loaded/saved</td>
<td>system</td>
</tr>
<tr>
<td>1</td>
<td>#<em>Day</em></td>
<td>7,6</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
<tr>
<td>1</td>
<td>#_GROUP_DIMENSION_DATA_REGIONS</td>
<td>0,6</td>
<td>loaded/saved</td>
<td>system</td>
</tr>
<tr>
<td>2</td>
<td>#<em>Dimensions</em></td>
<td>9,8</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
<tr>
<td>1</td>
<td>#_GROUP_DIMENSION_DATA_DATAWARE_HUB</td>
<td>0,8</td>
<td>loaded/saved</td>
<td>system</td>
</tr>
<tr>
<td>9</td>
<td>#<em>Measures</em></td>
<td>11,10</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
<tr>
<td>10</td>
<td>#_GROUP_DIMENSION_DATA_Measures</td>
<td>0,10</td>
<td>loaded/saved</td>
<td>system</td>
</tr>
<tr>
<td>11</td>
<td>#<em>Products</em></td>
<td>12,12</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
<tr>
<td>12</td>
<td>#_GROUP_DIMENSION_DATA_Products</td>
<td>0,12</td>
<td>loaded/saved</td>
<td>system</td>
</tr>
<tr>
<td>13</td>
<td>Sales</td>
<td>12,6,14,8,10</td>
<td>loaded/saved</td>
<td>system</td>
</tr>
<tr>
<td>14</td>
<td>#<em>CUBE</em></td>
<td>14,1</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
<tr>
<td>15</td>
<td>#_CONFIGURATION</td>
<td>16</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
<tr>
<td>16</td>
<td>#_DIMENSION</td>
<td>18,17</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
<tr>
<td>17</td>
<td>#<em>SUBSET</em></td>
<td>20,19</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
<tr>
<td>18</td>
<td>#<em>VIEW</em></td>
<td>22,21</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
<tr>
<td>19</td>
<td>#<em>SUBSET_LOCAL</em></td>
<td>17,18,19</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
<tr>
<td>20</td>
<td>#_SUBSET_GLOBAL</td>
<td>17,19</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
<tr>
<td>21</td>
<td>#<em>VIEW_LOCAL</em></td>
<td>14,16,21</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
<tr>
<td>22</td>
<td>#<em>VIEW_GLOBAL</em></td>
<td>1,21</td>
<td>loaded/saved</td>
<td>attributes</td>
</tr>
</tbody>
</table>
You can also find the API documentation on the website:

The access from other computers to the API is made possible in an analogous fashion as the access for Excel-Clients.

The entry

admin "192.168.2.100" 7778

would allow access over the URL http://192.168.2.100:7778/

The entry

admin "" 7778

would allow access over all IP-addresses of the computer, including 127.0.0.1 / localhost.

You can communicate with the Palo-Server by using the HTTP 1.1 protocol. You then receive the result in an easy-to-parse CSV display. You can also use the Legacy mode to maintain compatibility to Palo 1.0.
2.8 Backup of Databases

Palo comes with a demo database. Since modifications, which are performed while working further with this manual, are entered immediately into the database "Demo", it is wise to create a copy of this database and name it "Originaldemo". This is done as follows:

After the installation of Palo, you find the following symbol in the task list (system tray):

After double-clicking it, the following dialog window appears:

Click on "Stop". After a while, the colour of the "Start" button changes to green. At the bottom the following message appears: PALOServerService is stopped.

Now change the directory to the directory which contains the Palo database.

By default, this is the directory:

C:\Program Files\Jedox\Palo\Data

or:

C:\Documents and Settings\All Users\(Shared)Documents\Jedox\Palo\data.

As with an ordinary folder, create a copy of the database folder "Demo" and give it the name "Originaldemo".
Then click the button "Start" again. Depending on the number and sizes of the databases in the directory ...\Palo\Data, it will take the PALOServerService more or less time to update its service status. After some time, the colour of the "Stop" button changes to red. At the bottom the following message appears: PALOServerService is running.

Alternatively, you can do the following:

<table>
<thead>
<tr>
<th>With the right button of the mouse, click on this icon:</th>
<th><img src="image" alt="icon" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Then terminate the service by clicking on &quot;Stop Palo service&quot;.</td>
<td><img src="image" alt="icon" /></td>
</tr>
<tr>
<td>The icon changes to:</td>
<td><img src="image" alt="icon" /></td>
</tr>
</tbody>
</table>

Now change the directory to the directory which contains the Palo database. By default, this is the directory:

C:\Program Files\Jedox\Palo\Data

or:

C:\Documents and Settings\All Users\(Shared)\Documents\Jedox\Palo\data.

As with an ordinary folder, create a copy of the database folder "Demo" and give it the name "OriginalDemo".

<table>
<thead>
<tr>
<th>Click on this icon with the right button of the mouse:</th>
<th><img src="image" alt="icon" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>Start the service by clicking on the button &quot;Start Palo Service&quot;. The icon color will change immediately to green. But, depending on the number and sizes of the databases in the directory ...\Palo\Data, this will take more or less time to update the service status.</td>
<td><img src="image" alt="icon" /></td>
</tr>
</tbody>
</table>

**Tip:** Databases created by you are also stored in the folder ...\Palo\Data by Palo. The steps explained above equally apply to the databases created by you. You can copy them, rename them, create backups on various storage media or transfer them to other computers.

Please note: Do not use blanks in the name of the database folder! Otherwise, starting the Palo Service Manager will not work properly.
2.9 Deinstalling Palo

You deinstall Palo via Start\Control Panel\Add or Remove Programs:

You are asked to confirm when you click the REMOVE button. Confirm with YES if you really intend to remove Palo.

Then the actual deinstallation starts.
After deinstallation, either of the following messages will be displayed:

Or:

Further, the information displayed depends on the respective work environment.

A few files (data and log files) from the previously installed Palo version are not deinstalled. You must delete them manually after you restart your computer if necessary.
3 Palo Basics

3.1 How Palo Works
Palo was developed for Microsoft Excel. It is a cell-related in-memory database that is multidimensional and hierarchical. Palo also has an automatic attribute management. But what do these terms mean in detail?

3.2 Palo is cell-related
When you query a relational database, the smallest obtainable result is a data record (normally composed of several fields). When you start a query in Palo, the result is a single cell value. In other words, Palo thinks in terms of cells instead of data records and this is what makes Palo very compatible with Microsoft Excel since Excel is cell-related as well.

3.3 Palo is an in-memory database
Palo OLAP Server is an in-memory database, i.e. the database with which a Palo-user works is total loaded into memory. Therefore Palo is generally speed-oriented. Data queries to basic cells and sum total cells are performed in real time. There are no time-consuming queries, results should be shown in milliseconds (or seconds at the longest).
There are other multi-dimensional databases, based on relational technologies (ROLAP database), on the market. They have their advantages in terms of capacity, but they are mostly slower and do not normally support write-back of data to the cubes.

3.4 Palo is multi-dimensional
Multidimensionality is an alternative way of organizing data in a database. In a relational database you are used to structures such as tables, data records and fields. In a multidimensional database you are thinking in terms of cubes and dimensions and cells.

You know what a cell is, but what is a dimension and what is a cube? To answer the question it makes sense to take another look at Microsoft Excel.
In the next screenshot you find a list of row titles in Excel:

In Palo, this list is called a **dimension**. Basically, a dimension is a list of categories that has some contextual relationship like a list of products, regions, customers or months. In Palo the above Excel list would be modeled as a dimension e.g. with the name ‘Products’. So far in our Excel sample we have one dimension on the sheet. Now we are going to add another dimension: a horizontal list of months and quarters, which you can see in the figure below. This dimension is called “Months” in Palo.

Now there is an interesting point. What do you do in Excel if you want to arrange your data in more than two dimensions? You can add tables to generate a third dimension, e.g., by storing product names in tables and consolidating them into a product group in another table.

But what comes next? From experience we know that large organizations or companies generally require five to ten dimensions to display their data in line with their requirements. An Excel-worksheet only supports three dimensions.
This is one of the reasons why Palo was developed. Palo supports up to 256 dimensions. It allows to structure data in a much more detailed manner than Excel in its pure form.

To arrange dimensions, they are combined in a specific way, which results in the creation of **cubes**. Let’s take a look at such a cube.

As an example we will use our two-dimensional model that we developed in Excel. There is a cell range which is defined by the dimensions "Products" and "Months" (range C3:J22). It is composed of individual cells, and each of these cells has its own address (e.g., D4). This cell range is called a two-dimensional cube in Palo.

In principle, a cube is a collection of cells, which are defined by two or more dimensions. Now the question remains, how do you access and display data from a multidimensional cube if Excel is only capable of displaying two dimensions on a worksheet at a time?

To answer this question, we first need to take a look at the way a cell is referenced in Excel. Excel uses a simple naming scheme for its cells, it is called the A1 style. So the value for 'Desktop Pro' in February is stored in D4 (see above). In a Palo data cube the value would be stored in a cell that does not use letters and numbers as cell address, but the real-life description of the data instead. In our example the address of the cell would be 'Desktop Pro', 'Feb'.

A special database function called "**PALO.DATAC**" is used to display data from a Palo cube in Excel.

The following formula pulls a value from a multi-dimensional data cube into an Excel cell (ignore the server and cube names at this point):

\[ =\text{PALO.DATAC}("Servername", "Cubename", "Desktop Pro", "Feb") \]

The data formula may also contain more than two dimension coordinates, for example, to access a five-dimensional cube it might read as follows.

\[ =\text{PALO.DATAC}("Servername", "Cubename", "Desktop Pro", "Feb", "2006", "Europe", "Units") \]
In the following section, you can see how data from a six-dimensional cube are displayed in an Excel-worksheet using a six-dimensional data formula.

If you follow the references in the data formula (in the Formula Bar) you will discover that the data formula references the strings in the row and column titles to get some of the coordinates, but it also references cells which are on the top of the page (cell A3, A4, A5). These cells contain the coordinates for those dimensions that could not be displayed as row or column titles. Using these ‘page selectors’ is one way to display views on a multi-dimensional database in a two-dimensional spreadsheet grid.

This handbook has been personalized for NA - ohneanmeldung
Keep in mind that you only have to change the contents of cell A3 or A5 to see the data for a different region or year. As a result, you can display partial quantities of larger data stocks using only one single Excel-worksheet. Think of how many worksheets you would need if you want to display the same quantity of data using just Excel.
Finally, we want to explain how you enter values into a database. This is actually very easy. You simply overwrite the formula in a worksheet (e.g., in C11) with a value. The Palo Excel Add-in ensures that the value is automatically saved to the specific cell in the cube. After that, Palo restores the original formula, which then shows the new value.

If you enter values, it is important that you simply go to the corresponding cell, input the new value and then press "Enter". The PALO.DATAC-function transfers the value in the addressed cube cell and it must be preserved as a function in the cell. If you copy a value into the cell or delete the content of the cell with "Del", before you enter the data, the PALO.DATAC-function will be deleted. Thereby also deleting the connection between the Excel-cell and the addressed cube cell.
3.5 Working with Hierarchies

When looking at the last screenshot you might be wondering why the values for “Desktop Pro”, “Stationary PCs” and “All Products” all display the number 123 although we only entered the value for “Desktop Pro”. The answer to this question is hierarchical aggregation. The dimension “Products” is not just a basic list. When we created the dimension in the Palo database Modeller, we defined a hierarchy which looks like this:

![Hierarchical Dimension](image)

The use of hierarchies in defining a dimension increases the functional capacity of the Palo data cube. A cube can create multi-dimensional sums within a cube using hierarchical dimensions. Consequently, you do not perform these calculations using Excel formulas. Instead, you pull the aggregated values directly from the cube.
3.6 Working with Attributes

In addition to element names, you often need element attributes. Local peculiarities are a typical reason for using element attributes, i.e. there may be a need to display an element name depending on the language we are using. This information is in an additional table, which enables a more detailed and longer notation than an element name.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>_Months</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>German</td>
<td>English</td>
<td>French</td>
</tr>
<tr>
<td>5</td>
<td>Jan</td>
<td>Januar</td>
<td>January</td>
<td>janvier</td>
</tr>
<tr>
<td>6</td>
<td>Feb</td>
<td>Februar</td>
<td>February</td>
<td>février</td>
</tr>
<tr>
<td>7</td>
<td>Mar</td>
<td>März</td>
<td>March</td>
<td>mars</td>
</tr>
<tr>
<td>8</td>
<td>Apr</td>
<td>April</td>
<td>April</td>
<td>avril</td>
</tr>
<tr>
<td>9</td>
<td>May</td>
<td>Mai</td>
<td>May</td>
<td>mai</td>
</tr>
<tr>
<td>10</td>
<td>Jun</td>
<td>Juni</td>
<td>June</td>
<td>juin</td>
</tr>
<tr>
<td>11</td>
<td>Jul</td>
<td>Julii</td>
<td>July</td>
<td>juillet</td>
</tr>
<tr>
<td>12</td>
<td>Aug</td>
<td>August</td>
<td>August</td>
<td>août</td>
</tr>
<tr>
<td>13</td>
<td>Sep</td>
<td>September</td>
<td>September</td>
<td>septembre</td>
</tr>
<tr>
<td>14</td>
<td>Oct</td>
<td>Oktober</td>
<td>October</td>
<td>octobre</td>
</tr>
<tr>
<td>15</td>
<td>Nov</td>
<td>November</td>
<td>November</td>
<td>novembre</td>
</tr>
<tr>
<td>16</td>
<td>Dec</td>
<td>Dezember</td>
<td>December</td>
<td>décembre</td>
</tr>
</tbody>
</table>
4 First Steps: Creating a Palo Worksheet

Palo comes with a demo database. Therefore, you will not have to worry about database modeling at this point. Follow the instructions, and you will understand the steps while using the demo database.

Unless you defined that Palo should start automatically with Excel, start Palo by clicking the desktop icon or open it via Start/Programs/Palo/Palo Excel Add-in. When Excel opens, you will see a new menu item: "Palo".

4.1 Paste View

The first option in the Palo menu is the item "Paste View":

This dialog makes it possible to create Palo data views on a worksheet in a very simple way. Here you can select the dimensions and elements that you want to see and how they should be arranged.
If you are familiar with the pivot function in Excel, you will recognize some of the functionality. Dimensions can be moved around to be either row or column titles or to serve as selection criteria.

In the cube (here: Sales), the cube-type can now be set for data, attributes and users. An explanation of the various cube buttons follows:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Name of the symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="cube" /></td>
<td>Data cubes</td>
<td>This module enables access to standard cubes.</td>
</tr>
<tr>
<td><img src="image.png" alt="cube" /></td>
<td>Attribute cubes</td>
<td>This module enables access to attribute cubes.</td>
</tr>
<tr>
<td><img src="image.png" alt="person" /></td>
<td>User management cubes</td>
<td>The various rights of the different users and groups are stored here and it allows you to control write and read rights.</td>
</tr>
</tbody>
</table>
The following database functions can be used for the representation of data from a Palo cube:

- **PALO.DATA:**
  Reads/writes values on basis of the functions’ arguments, each cell gets calculated separately.

- **PALO.DATAC:**
  Reads/writes values on basis of the functions’ arguments, similar to PALO.DATA, but all DATAC formulas are calculated in one go. Other Palo functions will not work when they use results from a DATAC function.

- **PALO.DATAV:**
  Similar to PALO.DATA and PALO.DATAC, but the function works even faster because an array formula is generated for the complete area. Use of PALO.DATAV is only possible in connected cell areas.

With all three functions values of specific elements from the cube are retrieved. The difference between them is:

- PALO.DATAC is the standard and faster way to retrieve data.
- PALO.DATA is slower, but it can render values to other functions.
- PALO.DATAV is the fastest function, but it can only be used in connected cell areas and not if values are to be written. (As an array-function it controls several cube cells for display in Excel-cells simultaneously. Writing only works if it is contained in one single cell).

Move the dimension "Regions" to the Column titles area and "Products" to the Row titles area. The result should look like this:

As soon as you have moved a dimension to the row- or column-titles, you can double click "Select Elements…” for this dimension.
4.1.1 Dialog Window "Paste Elements"

After double-clicking "Select elements..." the dialog window "Paste Elements" appears. You can now select elements, which you would like to see in the result.

![Paste Elements Dialog Window]

### 4.1.1.1 Description of the Hierarchy Buttons

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Enable hierarchy view" /></td>
<td>Enable hierarchy view. This is a prerequisite for using the following buttons:</td>
</tr>
<tr>
<td><img src="image" alt="Open one hierarchy level" /></td>
<td>Open one hierarchy level</td>
</tr>
<tr>
<td><img src="image" alt="Open all hierarchy levels" /></td>
<td>Open all hierarchy levels</td>
</tr>
<tr>
<td><img src="image" alt="Close one hierarchy level" /></td>
<td>Close one hierarchy level</td>
</tr>
<tr>
<td><img src="image" alt="Close all hierarchy levels" /></td>
<td>Close all hierarchy levels</td>
</tr>
<tr>
<td><img src="image" alt="Enable list view" /></td>
<td>Enable list view</td>
</tr>
</tbody>
</table>
Click on "Open all hierarchy levels" in order to see all levels of the dimension.

You can select single elements and/or accumulated elements in any combination. There are various ways to select the elements which you would like to be displayed. You can select the desired elements individually in two ways:

- Mark the elements in the left window and then add them to the "Pick list" by clicking the blue arrow. To mark several elements, hold down the SHIFT and/or CTRL key.
- Doubleclick on the elements.

**Note:** In all further selections, the "add to Pick list"-step is displayed, because this offers further possibilities. So you can, for example, perform individual sorting in the "Pick list". Furthermore, with an extensive selection, you keep a better overview over what you have already selected. If however, you want to select only a few elements and you marked these in the left window, you can spare yourself this extra step. Simply click on OK after the selection.

Instead of this individual selection, you can also use the subsequently described selection tools in order to select a group of elements.
4.1.2 Selection Buttons for Element Selection

Description of the Selection Buttons:

<table>
<thead>
<tr>
<th>Show all selection tools</th>
<th>Select all</th>
<th>Select branch</th>
<th>Invert selection</th>
<th>Search &amp; Select</th>
</tr>
</thead>
</table>

Buttons for element selection.
1-5 represent the hierarchy levels
B selects base elements.

Level-selection:
Click on the selection button "1" in order to select all elements shown in the highest level (Europe). Click on , 2" in order to select all elements shown in the second level (West, East, ...) And so on. Or you click on "B" to select all base elements shown.
Select all:
Selects all elements shown. By clicking on the arrow which points to the right, all these elements are added to the "Pick list". There, they can be sorted according to your needs.

Select Branch:
This button selects the element and its child elements.
If the "Show all selection tools" checkbox is checked, two additional possibilities are shown:

**Invert selection:**
If, for example, you have selected all base elements by clicking on "B", clicking on "Invert selection" will make all non-base elements selected.

**Search & Select:**
This is a text search. You can either enter the complete name of an element or you can use wildcards before and/or after the text. For example, the pattern "*TT" matches "Server Power TT" as well as "Notebook TT".
You just got to know the different possibilities to select elements. Of course you can also sort the selected elements. In order to change the sequence, select an element in the window on the right side and then move it up or down with the blue arrows "Up" or "Down".

Alternatively, you can sort the selected elements in ascending or descending alphabetical order by clicking respectively on the buttons "Ascending" or "Descending".

In order to remove elements from the "Pick list", click on the button "Clear list".

After you have selected all positions which you want to be displayed, terminate the selection process by clicking on the "OK" button. If the "Pick list" is empty, clicking on "OK" will select all marked elements in the window on the left side.
4.1.3 Further Options in the Dialog-Window "Paste View"

Zero Suppression:
If "Zero suppression" in the "Paste View"-dialog is checked, the view for the year 2009, with "Europe" selected for Regions and "Select all" for Products, looks as follows:

Show element selector on double-click:
This activates the "Paste element" window when you double-click the first element of a row or column. If this selection is not activated, the elements are filled into the following rows.

With "Show element selector on double-click" Without this option, elements in the next lower level are also shown.
Indent:
If "Indent" in the "Paste View"-dialog is checked, the indented selection for "Products" looks as follows:

Wrap labels: Forces a line break, if the text in a column is too long.

Fixed width: Changes the width of a column – 14.00, by default.

Note: Line break and column width interact as follows: The line break is applied only to those lines, which are filled by using the Column titles. For these, the stated column width is used. Columns, which are filled by using the Row titles, are always displayed with the necessary width.

Yellow cells in the upper left corner represent the remaining dimensions of the current cube. These are neither rows nor columns, but they can be used as selection criteria. You can click on them, in order to select a different element, as, for example, a different year, a group of countries or a single country in Europe.
With the "Paste View"-command it is also easy to change the data-view. In the Palo menu, click on "Paste View", then drag "Datatypes" and "Months" into the Column titles area:

Now click on "Months" and then double-click on "Select elements...". The "Paste Elements"-dialog appears, select the months Jan, Feb and Mar.

Select "Budget" and "Actual" in the dimension "Datatype" in the same manner.
After you have finished the selection, you will see the following presentation:

On double-clicking, the row- and column-headers of a Palo-view offer additional functions.

- A double-click on a free area, on the left side (here: A7 and A8), opens the "Paste View"-dialog, where you can adapt the worksheet settings.
- Double-clicking on a consolidated element switches between hiding and showing elements of the subordinate level.

If "Show element selector on double-click" has been selected, double-clicking on the first element in a row or column will open the "Paste Elements"-menu.
Now you can select elements again:

**Tip**: We would like to direct your attention to a very important aspect of Palo. In order to make this visible, place the cursor in cell C10 and then press F2, in order to see the function and its references:

Cell C10 contains the function PALO.DATAC(). Its parameters are: Server/database and cube, as shown in cells A1 and A2, and the coordinates of the Datacube. In our example:

- $A9 = All Products
- $A$5 = Europe
- C$8 = Jan
- $A$3 = 2002
- C$9 = Actual
- $A$4 = Units
These parameters are sufficient, to enable the PALO.DATAC-function to address exactly one cell in the Sales cube. That is, to display values of the cube-cell in the Excel-cell or to write back values entered into the Excel-cell to the cube-cell. The PALO.DATAC-function is the most important function in Palo, because it connects Excel-cells with Palo-cells.

Note that the $-Symbol has already been inserted everywhere in a manner, which makes it possible to copy and insert functions.

Incidentally, the function refers to cell C7, which seems to be empty. That's only for optical reasons. If you click on cell C7, you will see that it contains the formula "=B7" and consequently the value "Jan", which is contained in the cell B7. This is achieved with the user-defined format ,, , which has the effect that the cell content will not be displayed.

### 4.2 Paste Elements

Instead of the "Paste view" function, you can insert element names into your worksheet using the "Paste elements" function. This especially makes sense if you want to individually structure data in the worksheet.

Create a new worksheet and make sure that the cursor is placed in the upper, left corner (A1). Click the "Paste elements" command in the Palo menu.
The selection tools and the logic correspond to those in the “Paste View” dialog, except that here you see all dimensions. Note that you can only handle elements from one dimension at a time.

If you select one single element, a “Paste” button appears. If you select several elements, you have the choice between “Paste horizontally” and “Paste vertically”. Let’s start to paste our page selector elements. With the “Paste View” function, these are the dimensions that are combined under “Page selector”. Open the dimension “Months” and confirm with “Paste”.

The window will close, and you will see in Excel that the cell A1 now displays “Year” and that a formula is contained.

Double-clicking on A1 opens a dialog window, which allows you to select a different element from the same dimension.
Use "+" and "-" to open and close one level. "Expand all" and "Collapse all" refers to the entire element tree.

Please note: When you search for an element via the "Search & Select" function, make sure that the element tree is expanded completely.

Place the cursor on A2, and open the "Paste Element" dialog again. Select the "Europe" element from the "Regions" dimension, and insert it in cell A2. Now we have two elements in the page selector section:

The next step will be to create the rows.
Place the cursor in cell B5, and open the "Paste Element" dialog again:
Open the “Products” dimension this time, and click “+” to the left of “All Products”. Then select the second level by clicking “2”. Then the following screen appears:

After you have selected the elements, click “Paste vertically” and you get the following:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Year</strong></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td><strong>Europe</strong></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><strong>Stationary PC’s</strong></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td><strong>Portable PC’s</strong></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><strong>Monitors</strong></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><strong>Peripherals</strong></td>
<td></td>
</tr>
</tbody>
</table>
To obtain a column title, place the cursor on cell C4. Open "Paste Element" again, and select "Budget" and "Actual" from the "Datatypes" dimension.

Then paste them horizontally.

Our worksheet should then look as follows:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Year</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Europe</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>Budget</td>
<td>Actual</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Stationary PC's</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Portable PC's</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Peripherals</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.3 Paste Data Function

Final step: We have to insert appropriate functions in order to call the data from the Palo-database. We use the "Paste Data Function"-command for it.

Mark the complete data area (C5:D8) in your worksheet, and select the "Paste Data Function" dialog. You must choose the database and the cube that contain your data. Make sure that "Guess Arguments" is marked, and then click "Paste".

Palo will now analyze the surrounding cells and compose a formula which makes sense in this context.

The information about "Server/database" and "Cube" must be provided by you. Palo sets the other parameters based on the data which you arranged on the worksheet.

When you select one of the newly inserted data cells (e.g., C5), you will see that the PALO.DATAC function fetches the correct data and inserts it in the worksheet. Because no elements from the dimensions "Year" and "Measures" are arranged on the worksheet, the first element of each ("2002" and "Units" respectively) is placed into the formula.

=PALO.DATAC('localhost/Demo','Sales',$B5,$A$2,$A$1, '2002',C$4,'Units')

This handbook has been personalized for NA - ohneanmeldung
4.4 Example of Paste Options

You have learned how you can create a worksheet manually using the functions "Paste elements" and "Paste Data Function". Of course, you can also expand previously created models with "Paste View" using these functions. Let’s try doing that now.

Insert the following in the worksheet with "Paste View":

Because you did not make a selection within "Product" and "Measures" via Select Elements..., the top level will be displayed respectively.

The following screen appears:

Because you did not make a selection within "Product" and "Measures" via Select Elements..., the top level will be displayed respectively.
Now click cell A10 in the worksheet. Under "Products" via "Paste elements" select the 2nd level and paste vertically.

The elements will be inserted below "All Products" and only the names of the elements are written into the cells which makes sense, this being a list of elements.

If only one single element is pasted on the other hand, only "Paste" is available instead of "Paste horizontally" or "Paste vertically". In this case, the element is inserted and you can change it with a double-click.

The view should then look as follows:

Continue by marking cell B9 in the "Units" column.
Continue in that you copy the function in cell B9 into the cells B10:B13. Accordingly, the function in cell C9 should be copied into the cells C10:C13.

The fields will be filled with values automatically (a few of the results are negative, because the data type is "Variance", i.e. a deviation).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo Sales</td>
<td>Units</td>
<td>Gross Profit</td>
</tr>
<tr>
<td>2</td>
<td>Europe Year</td>
<td>29,335.00</td>
<td>3,495.21</td>
</tr>
<tr>
<td>3</td>
<td>2002</td>
<td>2,181.60</td>
<td>18,788.97</td>
</tr>
<tr>
<td>4</td>
<td>Portable PC's</td>
<td>-13,043.68</td>
<td>-60,768.94</td>
</tr>
<tr>
<td>5</td>
<td>Monitors</td>
<td>36,480.28</td>
<td>28,400.39</td>
</tr>
<tr>
<td>6</td>
<td>Peripherals</td>
<td>1,730.60</td>
<td>7,076.78</td>
</tr>
</tbody>
</table>

4.5 Save as Snapshot
This function saves the current version of the Excel-worksheet including the values, as a matter of fact, the Palo formulas are replaced by values.
"Save As Snapshot" creates a "copy" of the current data view.

This feature makes it possible to forward the worksheet.

An example is shown below.

The original contains PALO-functions, e.g. in C13

Save as snapshot can be very useful, for example, if data is to be forwarded to a specific distribution list which may include recipients who do not have access to the Palo database (e.g., external offices)
4.6 Work Offline

This menu is used to see data of the Palo database without a connection to the server. Therefore you have to turn on a client cache.

You get the dialog window "Cache Settings" with "Palo - Modeller" and then the switch "Database Settings".

Then, you create views of the data which you want to see offline. After switching to "Work Offline" you can then see the data although offline. However, if you seek further data or details, you will get the message "#null". This is because these data were not transferred to the client cache during the status was online:
5 Working with the Modeller: Developing Data Structures

5.1 The Modeller

Data structures are created from cubes, dimensions and elements. We will now demonstrate how to create and manage cubes, dimensions and elements with the Modeller.

When you start "Palo – Modeller", you see the following screen:

You might think that you must first create a cube and then define its dimensions. But we will see that it is more logical to first create the dimensions and then define one or more cubes, which can contain different combinations of dimensions.

Before we create cubes, dimensions and elements, we should take a look at the set of allowed characters in Palo.
5.2 Allowed and not allowed characters in Palo

- Characters allowed in **database-names**: 
  - ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789_-.
  - Furthermore, the name cannot begin with a point.

- Characters allowed in **element-names**:
  - ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789_-.
  
  The name of an element has to obey the following rules:
  - The length cannot be 0.
  - It cannot begin with a blank.
  - It cannot end with a blank.
  - It cannot contain any characters with character code < 32, e.g.. Return, Tab etc..

- Characters allowed in **dimension-names**:
  - ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789_-.
  
  The name of a dimension must obey the rules for element-names. **Additionally**, it has to obey the following rules:
  - The name cannot begin with a point.
  - The name cannot contain any of the characters \ / ? * : | < > and it cannot contain blanks.

- Characters allowed in **cube-names**:
  - ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789_-.
  
  The name of a cube must obey the rules for element-names. **Additionally**, it has to obey the following rules:
  - The name cannot begin with a point.
  - The name cannot contain any of the characters \ / ? * : | < > and it cannot contain blanks.

**Please note**: Names in Palo are case sensitive!
5.3 Creating Dimensions

Let us add a new dimension ‘Customers’ in the database "Demo". Please click in the area dimensions of the Database tab and then press "Enter":

The entry "New Dimension" is created and appears immediately as a new tab (tab) at the top of the window. Then enter the name "Customers" for the new dimension and finish this entry with "ENTER":

Switches to sort the dimensions and the elements:

- Sorts in order of creation
- Sorts alphabetically in ascending order
- Sorts alphabetically in descending order
Note: Theoretically, you can create up to 256 dimensions in Palo and also build cubes with them.

In practice, however, there are two limitations:

1. Due to the number of function arguments, which Excel can process. (e.g., Excel 2003 can process 30 function arguments, that is, due to PALO.DATA(database-name, cube-name, dimensions) we’re limited to 28 dimensions)

2. Performance-problems, if the possibilities of processor and RAM are overtaxed by the number of addressed cube cells.
With the button "New Dimension" in the lower left corner of the dialog window you will get the dimension wizard to use a dimension from the templates. This has the advantage that the new dimension will have all the elements of the selected template.

We can enter e.g. as the name "Key_Performance_Indicators" and select the template "Measures":
As a result we get:

The elements of the template dimensions "Measures" are inclusive.

### 5.4 Creating Elements

To add elements for the new dimension "Customers" you can either double-click the entry "Customers" in the left part of window or click on the tab at the top of the window. In both cases, you get a new window that allows you to add elements to the dimension "Customers":

Press "Return", in order to create the first element. Create four elements: "Smith Corp", "Meyer Ltd", "Miller LLC" and "Total".

After you pressed ENTER for the first time, a new element was created. It is marked and can now be given an adequate name. After the name has been entered, you have to press ENTER again, for confirmation.

If you press ENTER twice consecutively, the new element with the name "Smith Corp" (in our example) is completed and another new element is created immediately.

With the blue arrows, you can move elements within the dimension. The ordering of the elements can be chosen freely, but we recommend ordering these hierarchically in order to have a better overview over the data.
We will combine the elements in "Customers" into "Total" in the next step, i.e. turn "Total" into a consolidated element.

Start by double-clicking "Total", which then appears in bold text. You can now see the text in the window on the right: "Click an element from the list to add it.", i.e. you have to define which sub-elements belong to "Total". You can either allocate the elements with a double-click or by copying the items to the right using the blue arrow pointing to the right. Select the three elements previously created.

Click the green "OK" button at the lower window margin to finish.

In the figure below, you can see that the icons to the left of the elements are different.

Click the green "OK" button at the lower window margin to finish. In the figure below, you can see that the icons to the left of the elements are different.

Please note: Elements can be added to more than one consolidated element. This makes different views of a dimension possible - these are often called "parallel hierarchies".

This handbook has been personalized for NA - ohneanmeldung
5.5 Display of Hierarchies

In order to show the hierarchies, click on the hierarchy-button (red arrow).

You get the following view:

The description of the hierarchy-buttons can be found in chapter 4.1

The Modeller provides additional functions when you right-click the element (hierarchy view must be turned off):

- Add, delete, rename and consolidate
- Switch the type between numeric and text
- Copy & paste elements, select all
- Move elements to the start/end
- Count elements
- Search for elements
5.6 Copying and Inserting Elements

If you want to add elements to your dimension, you can simply write them into the worksheet. You can also use the “Fill” function in Excel and copy the items by using the context menu (copy and paste).

1. Create elements in Excel
2. Copy from Excel
3. Paste in Palo
4. The elements have been inserted

In above example, we added individual elements to a dimension. In this case, the elements are added to an existing dimension. You will see later that it is also possible to transfer structures in addition to elements. The structures determine the allocation to a dimension. But you will first learn how you copy elements from the Modeller into a worksheet.
We copy all elements of "Months" into an excel-shee:

![Excel Spreadsheet Screenshot]

We are moving to the spreadsheet and paste the contents of the clipboard into A1. We get the following:

The element names, element types and consolidation factors are copied. Element types are: "N" = numeric, "S" = string/text and "C" = consolidated.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>N</td>
<td>Jan</td>
</tr>
<tr>
<td>2</td>
<td>N</td>
<td>Feb</td>
</tr>
<tr>
<td>3</td>
<td>N</td>
<td>Mar</td>
</tr>
<tr>
<td>4</td>
<td>N</td>
<td>Apr</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
<td>May</td>
</tr>
<tr>
<td>6</td>
<td>N</td>
<td>Jun</td>
</tr>
<tr>
<td>7</td>
<td>N</td>
<td>Jul</td>
</tr>
<tr>
<td>8</td>
<td>N</td>
<td>Aug</td>
</tr>
<tr>
<td>9</td>
<td>N</td>
<td>Sep</td>
</tr>
<tr>
<td>10</td>
<td>N</td>
<td>Oct</td>
</tr>
<tr>
<td>11</td>
<td>N</td>
<td>Nov</td>
</tr>
<tr>
<td>12</td>
<td>N</td>
<td>Dec</td>
</tr>
<tr>
<td>13</td>
<td>C</td>
<td>Qtr. 1</td>
</tr>
<tr>
<td>14</td>
<td>Jan</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>Feb</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>Mar</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>C</td>
<td>Qtr. 2</td>
</tr>
<tr>
<td>18</td>
<td>Apr</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>May</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>Jun</td>
<td>1</td>
</tr>
<tr>
<td>21</td>
<td>C</td>
<td>Qtr. 3</td>
</tr>
<tr>
<td>22</td>
<td>Jul</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>Aug</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>Sep</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>C</td>
<td>Qtr. 4</td>
</tr>
<tr>
<td>26</td>
<td>Oct</td>
<td>1</td>
</tr>
<tr>
<td>27</td>
<td>Nov</td>
<td>1</td>
</tr>
<tr>
<td>28</td>
<td>Dec</td>
<td>1</td>
</tr>
<tr>
<td>29</td>
<td>C</td>
<td>Year</td>
</tr>
<tr>
<td>30</td>
<td>Qtr. 1</td>
<td>1</td>
</tr>
<tr>
<td>31</td>
<td>Qtr. 2</td>
<td>1</td>
</tr>
<tr>
<td>32</td>
<td>Qtr. 3</td>
<td>1</td>
</tr>
<tr>
<td>33</td>
<td>Qtr. 4</td>
<td>1</td>
</tr>
</tbody>
</table>
You can also do this in reverse order. Assume that you want to add additional customers to the “Customers” dimension. These should be combined into a group at the same time. In other words, you must supply two pieces of information:

1. The customer as base elements and element type.
2. Indication of the customer group that the customers belong to.

**Note:** We explained that elements will be added to existing structures when copied individually. This is not the case with a three-column copy - here all existing elements within a dimension will be deleted and replaced by the copied elements. Structures are adapted and set up in Palo as indicated in the columns in Excel. Consequently, this copying option is especially suitable if you want to add elements to a new dimension. For example, this could be the case when transferring from other MOLAP databases since they use the same arrangement format.

At the beginning of this chapter, we created the dimension “Customers” containing the names of your customers:

![Excel screenshot showing the dimensions and customers]

Now we want to add customers to this dimension.

Enter in a blank Excel-worksheet the data displayed on the right:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
<td>Group A</td>
</tr>
<tr>
<td>2</td>
<td>Customer1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Customer2</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Customer3</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>Customer4</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Customer5</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>Customer6</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>N</td>
<td>Customer1</td>
</tr>
<tr>
<td>9</td>
<td>N</td>
<td>Customer2</td>
</tr>
<tr>
<td>10</td>
<td>N</td>
<td>Customer3</td>
</tr>
<tr>
<td>11</td>
<td>N</td>
<td>Customer4</td>
</tr>
<tr>
<td>12</td>
<td>N</td>
<td>Customer5</td>
</tr>
<tr>
<td>13</td>
<td>N</td>
<td>Customer6</td>
</tr>
<tr>
<td>14</td>
<td>N</td>
<td>Customer6</td>
</tr>
</tbody>
</table>

All customers are numeric (“N”), consequently they can be added to a group. Within the group, all customers have the same weight, the consolidation-factor = 1. The feature C for “Group A”, combines the customers accordingly.

Select the area (here: A1:C14), and copy the selected area.
Now open the Modeller, and go to the "Customers" dimension. Select any element. Then use the context menu in Palo, and click "Paste Element".

A warning is displayed that all existing elements will be deleted if you continue the copying procedure. Confirm the message with "Yes". The elements are then deleted and at the same time, the new elements are added.

Pay attention to how the customers are combined within "Group A". It reflects the columns in Excel which specified the structure of the dimension.
The feature C for "GruppeA" combines the customers accordingly.

You see that Palo has got the following information through the application of the "Paste Elements"-function to the Excel-range A1:C14: Name of the element, C- or N-element and the weighting factor.

The command "Count" in the context menu counts the elements and returns the number for "n" = numeric, "s" = string/text and "c" = consolidated.
The Palo context menu provides further options. One of them is the ability to search for elements. Click “Search for element” to start the search. A window will open.

For example, if you enter “Denmark”, Palo jumps to an element in the current dimension, which matches the search criteria.

You must click on any element in the dimension in order to be able to select the search function in the context menu.
Searching with placeholders is possible as well. For example, entry of "Ju*" results in the hits "Jun" or "Jul". But only the element found first is selected.

You can also use "Ctrl + F" to call the dialog box "Search Elements". After you have found the first element you can continue the search with "F3".

Note: If no matching element exists, an error message is displayed.
5.7 Working with Attributes

Attributes are additional element properties. With attributes you can store any number of additional element properties similar to a table. The use of attributes can make sense, for example, if you want to display element names in different languages. The attribute enables you to display the respective element name in the selected language.

5.7.1 Creating Attributes

Attributes were already mentioned in Chapter "Palo Basics". Now you will create these attributes and fill them accordingly. To use attributes, open the Modeller, click the dimension "Months" and activate "Attributes".

1. Activate "Attributes"

Add the three elements "German", "English" and "French". You learned how to do that when creating the dimension "Customers" and its elements. Alternatively, you can also proceed here as follows. Click "New element" in the bottom left corner. Name the element, and repeat the procedure.

Once the elements have been created, they must be changed to "Text" using the context menu.

The setting "Numeric" can be more appropriate for other purposes.

Please note: You must select and change each element individually.

**Note:** Only attributes with the string setting will allow you to retrieve the replacement names, i.e. in this example the translations of original element names. We will deal with this difference again in the "Searching for Attributes" section.
The result with the "string" setting appears as follows:

Go to "Paste View", and click the "attribute cube" icon in the middle.

You have just created the attribute cube "#_Months". The other attribute cube, "#_Products", will be required later. Select the attribute cube "#_Months.

Now you can change the settings for the view:

When you double-click "Select elements…" of "#_Month" in the Column titles, you are able to select the attributes that you want to insert.
Select the three attributes shown above.

Then repeat the procedure for "Months". With the button "Open all hierarchies" (red arrow), all elements will be shown:
Then select the base months.
Click "B" to do that.

After you have selected the base months, they can be transferred into the Pick list by clicking the arrow. But you could also click OK directly, because you have already selected all base elements.

The first alternative makes sense, for example, if you want to change the order of the elements or if you do not want to transfer all of the base elements.
With “OK” and “Paste” you obtain the following view:

As shown on the left, the cube has not yet been filled. Manually enter the content of the Excel table (as shown on the right) into the Palo database table.

The word “manually” is important here: After every input-confirmation, the entry is written into the corresponding cube cell and is shown again by the Palo-function in the corresponding cell. Therefore, make sure that, after editing, the Palo-function is still displayed in the Formula Bar (If you copy entries into cells, this function would be overwritten! - If this happens, you should correct the inadvertent deletion with the Excel-command "Edit – Undo")

The result should look as follows:

<table>
<thead>
<tr>
<th>Month</th>
<th>German</th>
<th>English</th>
<th>French</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>Januar</td>
<td>January</td>
<td>janvier</td>
</tr>
<tr>
<td>Feb</td>
<td>Februar</td>
<td>February</td>
<td>février</td>
</tr>
<tr>
<td>Mar</td>
<td>März</td>
<td>March</td>
<td>mars</td>
</tr>
<tr>
<td>Apr</td>
<td>April</td>
<td>April</td>
<td>avril</td>
</tr>
<tr>
<td>May</td>
<td>Mai</td>
<td>May</td>
<td>mai</td>
</tr>
<tr>
<td>Jun</td>
<td>Juni</td>
<td>June</td>
<td>juin</td>
</tr>
<tr>
<td>Jul</td>
<td>Juli</td>
<td>July</td>
<td>juillet</td>
</tr>
<tr>
<td>Aug</td>
<td>August</td>
<td>August</td>
<td>août</td>
</tr>
<tr>
<td>Sep</td>
<td>September</td>
<td>September</td>
<td>septembre</td>
</tr>
<tr>
<td>Oct</td>
<td>Oktober</td>
<td>October</td>
<td>octobre</td>
</tr>
<tr>
<td>Nov</td>
<td>November</td>
<td>November</td>
<td>novembre</td>
</tr>
<tr>
<td>Dec</td>
<td>Dezember</td>
<td>December</td>
<td>décembre</td>
</tr>
</tbody>
</table>

This handbook has been personalized for NA - ohneanmeldung
5.7.2 Searching for Attributes

Now that you have entered the attributes for the "Months" dimension, they can be utilized.

You learned in the "Creating Worksheets" chapter that you can search for elements. Of course, this search can also be performed via attributes.

But the settings must be adapted. You can call the "Choose Elements" menu via the "Paste view" dialog. You access the same window directly via "Paste elements". Consequently, the procedure for both ways is the same.

In an empty worksheet, let's place the cell cursor in cell A1 and then we choose "Palo – Paste elements".

In this "Paste Elements"-window select "Show all selection tools".

In the bottom left corner of the "Paste Elements" window, you find a list box which is blank initially. As long as this is the case, you can only search for element names (e.g. Jan, Feb).

Above: "Paste Elements" without setting for attributes (list field = blank)
Below: "Paste Elements" with setting for attributes (list field = "English")

If you want to search for attributes, three prerequisites must be fulfilled:
1. Attributes must be maintained for the element.
2. You must activate these attributes. Set the list box to any attribute (e.g., German) in the example "Year".
3. The attributes must exist as text.

If you open all hierarchies, enter "march" in the search field and then press the "Search & Select"-button, the element "march" will be found (red arrow).

Then click on "Insert"
Then the "Choose Element for Months" window changes. Depending on the attribute language setting, you will be searching for January, for example. On the other hand, if you want to search for real element names, you must set the list box to "none".

The Palo formula also changes in accordance with this logic.
In the example "Year" and <none>, A1 contains:
=PALO.ENAME("localhost/Demo","Months","Year",1,"")
In the example "Year" and "German", A1 contains:
=PALO.ENAME("localhost/Demo","Months","Year",1,"","German","Sheet1!__palo622272")

The search for attributes is only possible when they exist in text format.

Note: The attribute description in the Palo.Ename-function is not editable. The past parameter "Sheet1!..." is set by the system and cannot be changed.
You have read in the "Creating Attributes" section that an attribute cube "#_Products" exists in addition to the attributes you just created.

It contains the product prices - consequently conventional, numeric data.

If you double-click in the Row titles on "Select Elements..." below "Products", then you see the following structure:

On the left side you see that no alias-selection is possible for "Products".

The reason is shown in the Modeller:
The attribute "Price per unit" is numeric:
You see here that the selection does not work for numeric attributes. For comparison, we temporarily change the attribute property to "Text".

The dimension "Products" in the Modeller – the attribute property has been changed to "Text".

Now "Alias" can be altered.

You just saw that the list box only works with text attributes. There is one more special feature to note in this context. As soon as you use a consolidated attribute element with the name "Alias", only those associated consolidated elements (child elements) are displayed in the list box.

For example, the term "alias" could combine the various language versions into one group. But you can also use a different generic term, e.g., "Languages". This means that "Languages" is not displayed as a generic term either. And this makes sense, because the purpose of this list box is to get the result in a specific language. Thus the display of the term "Languages" is not required as a selection criterion.

But the term "Alias" has further connotations with respect to the display. To understand these differences better, create another attribute in the "Products" dimension using the Modeller. Depending on whether Alias exists as a consolidated element, we distinguish a total of four variants. Let's look at our first example.
Enter the attributes German, English, Product Manager, Color and Alias. Alias consolidates the attributes German and English.

Now insert this information into a blank Excel-worksheet. Call "Paste View", and select the attribute cube "#_Products".
Select the attributes Productmanager, Color, English and German in the Column titles. Select the products in the Row titles as shown in the following screenshot:

Then the result appears as follows:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo</td>
<td>#_Products</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Desktop L</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Desktop Pro</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Desktop Pro XL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Notebook SX</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Notebook GT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Notebook LXC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
You can see that this overview is still blank. Enter the information as per the screenshot below.

![Excel spreadsheet screenshot]

You provided additional information for the products with these attributes. The element "Desktop L" now has the information "Color = Black" and "Product Manager = Mayer". In addition, in the English speaking countries, this desktop is sold under the name "Desktop Start" and in the German speaking countries under the name "Desktop Einsteiger".

When selecting information, it is important to know what you want to achieve with the selection. If the purpose is to display alternative names only, the provision of additional options does not make sense. "Alias" limits the selection to the required display of "German" and "English".

List box display variants:

**Variant 1**
Alias is found as a consolidated element and consequently only German and English are displayed in the list box in "Paste View/Choose Element" or "Paste Element".

**Variant 2**
If no consolidated element with the name "Alias" exists, all base elements are displayed in the list box in "Paste View/Choose Element" or "Paste Element".
Before the selection

After the selection "German"

Call the Modeller. Delete "Alias" via the "Delete element(s)" button.

Create instead "Other attributes", and consolidate "Product Manager" and "Color" here.
If you now open the "Paste Elements"-dialog, you see the following view (after Excel has been restarted):

To variant 2:
"Alias" was not found as a consolidated element. Consequently, German, English, Product Manager and color are displayed. "Other attributes", being a consolidated element, is not displayed.

Now you can display all products that, for example, Product Manager "Müller" is responsible for. Insert the cube "#_Products" in the "Paste View" dialog, and select the elements "German" and "Color" in the Column titles. Select "Portable PC's" in the Row titles, and set the list box to "Product Manager". Once you have the view in Excel, double-click "Portable PC's".
The following table is the result:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>localhost/Demo_</td>
<td>#_Products</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Portable PCs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Müller</td>
<td>Notebook Einsteiger</td>
<td>silver</td>
</tr>
<tr>
<td>7</td>
<td>Müller</td>
<td>Notebook Spiele</td>
<td>blue</td>
</tr>
<tr>
<td>8</td>
<td>Müller</td>
<td>Notebook Büro</td>
<td>black</td>
</tr>
<tr>
<td>9</td>
<td>Notebook TT</td>
<td>silver</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Notebook SL</td>
<td>blue</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Subnote SL</td>
<td>black</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Subnote XK</td>
<td>silver</td>
<td></td>
</tr>
</tbody>
</table>

This screen demonstrates two things:

1. The first column displays the Portable PCs which are allocated to a product manager. If there is no allocation, the original name is used as an alternative.
2. Column “E” follows column “C”. In other words, one column is hidden in between. It contains the original names for the attributes. Display this column by selecting the columns C:E and then Format/Column/Unhide.

You will see the following:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>localhost/Demo_</td>
<td>#_Products</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Portable PCs</td>
<td></td>
<td>Portable PCs</td>
</tr>
<tr>
<td>6</td>
<td>Müller</td>
<td>Notebook Einsteiger</td>
<td>Notebook SX</td>
</tr>
<tr>
<td>7</td>
<td>Müller</td>
<td>Notebook Spiele</td>
<td>Notebook GT</td>
</tr>
<tr>
<td>8</td>
<td>Müller</td>
<td>Notebook Büro</td>
<td>Notebook LXC</td>
</tr>
<tr>
<td>9</td>
<td>Notebook TT</td>
<td>Notebook TT</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Notebook SL</td>
<td>Notebook SL</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Subnote SL</td>
<td>Subnote SL</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Subnote XK</td>
<td>Subnote XK</td>
<td></td>
</tr>
</tbody>
</table>

This means that column “D” contains the original names of the products, which correspond with the respective attributes. Thus, if you change the Excel view, you can display the original names and the employees responsible for the products next to each other.
If no attributes (here: Product Manager) are maintained, for example, as for "Notebook TT", the missing attributes are replaced by the original name in column "A". You can also see the same in the "Paste Element" dialog:

Call the Modeller. Delete the consolidated attribute "Other attributes". Create an attribute with the same name again. Do **not** consolidate it!
Your entry of the individual attributes should look like this:

In the Modeller:

Variant 3:
No "Alias" was found, and there is no other consolidated element either.
"Other attributes" is a normal text attribute here. Result: all attributes are displayed.

You get the same results as in variant 2 with this selection. The attribute "Other attributes" does not result in a display different than "none". Reason: There are no entries in "Other attributes". Consequently, no alternative displays are possible.
Call the Modeller once again. Change the attribute type of "Other attributes" from "Text" to "Numeric". Consolidate "Color" and "Product Manager" in it again. Create another element with the name "Language". Consolidate "German" and "English" in it.

Consolidation in the Modeller

The consolidations now correspond to:

Language = Alias from variant 1
Other attributes = variant 2

Variant 4:
Alias was not found as a consolidated element. Consequently, German, English, Product Manager and color are displayed. "Language" and "Other attributes" are consolidated elements. Consequently, they are not displayed.

Summary: You can control via the structure of attributes whether there is a search for specific attributes or whether the search option is suppressed by not displaying the attributes in the list box. The term "Alias" will limit the display the most.
5.7.3 Information Processing with Attributes

Of course, you can not only search by attributes, you can use the feature to display results. Proceed as follows: Use "Paste View", change the cube to attribute cube and select the attribute "German" under "#_Month".

Check "Show element selection on double-click", and then call "Months". Choose the settings to get the following screen: Note that you can now select the attributes in the list box at the bottom.
Now double-click “German” in cell B4. A selection menu opens. You can now set a different language. The data are converted immediately.

You just learned how you can use attributes, e.g., for language control. But you can also use them to display numbers which you will see by looking at the attributes stored in unit prices. However, these attributes are not suitable as search criteria.

Display the product prices next. To do this, call the attribute cube "#_Products" via "Paste View", and display the prices of the first five products.
5.8 Creating New Cubes

Now we want to create a new cube. Start the Modeller, and click "Database". Then click the icon "New cube".

This will start the Cube Creation Wizard. Enter a name for the cube, e.g., "Market". Add the dimensions Datatypes, Months, Measures, Products, Regions and Years by double-clicking them, or use the blue arrow button. When you click "Finish", the cube will be created.

We want to use "Market" later to demonstrate additional features, therefore please create another new cube and name it "Test". This cube will be needed later to demonstrate deletion of a cube. After having created "Test", you will see the various cubes in the Modeller.
You have created two additional cubes. You can use them the same way as the pre-fabricated cube “Sales” from previous examples.

For example, you can retrieve information from the cube "Market" via “Paste View”. In "Paste View", you will notice the newly created cubes "Market" and "Test" in addition to the already familiar cube "Sales".
You will see that the newly created cubes "Market" and "Test" do not yet contain any data (cf. the figure below from the cube "Market"). You might wonder why? After all, both cubes were created by using the same dimensions that already existed in the cube “Sales”.

![Microsoft Excel - Book1](image)

You might think that these are copies of “Sales” and consequently should contain data. This is not the case, instead, the two new cubes contain new, independent data cells.

You may remember that we stated in the beginning of this chapter that you should first create the dimensions and then a cube. It is important to note that the Palo formula contains a reference to their respective cubes.

**Please note:** The sequences of the dimensions in the cubes "Sales" and "Market" are different. This sequence is essential for addressing a cube cell!
Cell B9 contains the following formula: \( \text{PALO.DATAC($A$1,$A$2,$A$9,$A$3,$A$4,$A$5,$A$6,B$8)} \).

If you convert the cell addresses into text, you get: \( \text{PALO.DATAC(“localhost/Demo”, “Sales”, “All Products”, “Europe”, “Year”, “2002”, “Variance”, “Units”)}. \) Consequently, the dimensions are allocated precisely to the respective cube, e.g., “Sales”. The sequence of the dimensions in the formula corresponds to the sequence in the cube. Compare the PALO.DATAC formula above with the figure in the upper right.

If you use these very same dimensions for a new cube, the new cube is empty as a result. It therefore makes sense to create the cube only after you know the dimensions.

Reason: the cube is a combination of dimensions. A dimension is in turn a combination of individual elements. Consequently, a cube cannot be described unless we are aware of this structural information.

Of course, you could start to fill a new cube by entering data into the cells. But generally, the data exist already in an external data source. In this example, they are actually in the database, just in another cube. Furthermore, it would be very time-consuming and a great error-source if you want to enter manually all data for all products, regions and years.

Consequently, we must now think about a way of filling a new cube with data that already exists. We will deal with this in the “Importing and Exporting Data” chapter. But first we want to deal with deleting cubes.
5.9 Deleting Cubes

You just learned how you create a new cube. You learned that the structure of elements and dimensions is only valid within a cube. For that reason, no data exist in the new cubes “Market” and “Text”, although they are structured the same way as the cube “Sales”.

Inversely, this also means that only the data and structures of this cube are deleted if you delete an entire cube. If the same dimensions and possibly data also exist in other cubes, they will be retained in those cubes.

You can test this by deleting the cube “Test”. Proceed as follows: Insert a view of the cube “Test” in a blank worksheet via “Paste View”. Place “Products” in the Row titles, whereby you select “Desktop L” as your product. Proceed in the same way for the Column titles, whereby you use the dimension “Datatypes” and the element “Actual”. The other dimensions remain in the Page selector. Then the following view appears:

![Microsoft Excel - Book1](image)

Make the following adjustments to the view:

- "Europe" becomes "Germany" (A5)
- "Year" becomes "Jan" (A3)
- The value "500" is entered for "Germany" in January 2002 (B9).

Then you see the following result:

![Microsoft Excel - Book1](image)

Undo the settings for "Germany" and "Jan".
You know that the same structures also exist in the cube "Sales". Place these data sheets next to each other for comparison. Create a view of the cube "Sales" in a blank Excel-worksheet using "Paste View". Please put "Products" in the Row titles as well, whereby you select "Desktop L" for your product. Proceed in the same way for the Column titles, whereby you use the dimension "Datatype" and the element "Actual". The other dimensions remain in the Page selector.

The results appear as follows:

Once again, you will recognize that the cubes have completely different contents although their dimensions are identical. Consequently, you can delete the cube "Test" without interfering with the cube "Sales". The complete structure of the "Test" cube is erased when deleting the cube.

Select the cube to be deleted in the Modeller. Call the context menu with a right-click, then select "Delete cube…".

This handbook has been personalized for NA - ohneanmeldung
Alternatively, you can click the "Delete cube..." button at the lower window margin. Make sure to select the correct cube, because the data are deleted irretrievably. Confirm your selection by clicking "Yes".

5.10 Additional Techniques in Working with the Modeller

You can create and manage not only cubes, dimensions and elements, hierarchies and attributes with the Modeller. You can also enter additional comments or you can combine elements with calculations.

5.10.1 Comments

Assume you want to add comments to your cube "Sales" to explain data elements in more detail, for example. Open the Modeller, and create an element "Comment" within the dimension "Months". Convert it into a string element using the context menu:

1. Create "Comment"
2. Convert into a string
Create an Excel view via "Paste View".

Select from "Months" some months and the new element "Comment". Confirm the selection with "OK". Proceed analogously for "Products".
Then the following screen is displayed:

5.10.2 Display of Long Comments

Up version MS Excel 2000 an Excel cell is limited to 255 characters. For reasons of compatibility with MS Excel 2000 and earlier versions Palo has retained this limitation, i.e. Palo functions display 255 characters in an Excel cell at a maximum. If you want to write a comment longer than 255 characters, you can do the following: Double-click in cell E9 opens the following window. It allows you to write a text at any length. You can later display the text with the array-function PALO.DATAT(). Copy now just this text and try it (549 characters):
If you format cell E9 with "wrap text", you get this view:

You can see that only the first 255 characters are displayed in E9.

Use the **PALO.DATAT** function to display the complete information. This function outputs a number of strings, each of which is not longer than 255 characters. Execute the function as follows: Insert two blank rows below the cell in question (here E9).

This way, you get two cells, which can show the continuation of the comment. Copy E9 to E10 and to E11. At first, these are empty:
Now, select the three cells E9, E10 and E11. **Important:** The selection must absolutely be in the sequence E9, E10 and E11. Then $A10$ automatically becomes $A9$. If you select E11 to E9, you will get three blank fields. Reason: In this case, $A9$ is changed to $A10$. The adjustment is done automatically in accordance with the sequence of selecting the cells.

Click behind the function in the Formula Bar, and finish the input with simultaneous pressing **STRG+SHIFT+ENTER** (necessary to enable the array-functionality). The result can be seen in the screenshot below:

![Excel screenshot](image-url)

This handbook has been personalized for NA - ohneanmeldung
You have now generated an array. You can recognize the array by the curved brackets (braces). You cannot simply make changes in this array, e.g., to the formula. In such an array-function, you can perform only alterations, if the entire cell-area, in which such an array-function was input, is marked again.

If you want to cancel the array-function, proceed as follows:
Mark the three cells E9, E10 and E11 again. Click with the mouse on the Formula Bar and then press the key-combination STRG+CAPS LOCK+ENTER: The array-functionality will be turned off.

5.10.3 Calculated Elements (Push Rules)
One way to work with calculated elements is to set up so-called server rules in the database. Computations are done within the database – complex models with large data volumes have an impact on speed as literally millions of operations may be required to calculate the entire cube. These rules are available since Palo Version 2.0 and are explained in detail in chapter "Enterprise Rules".

A good alternative is the use of push rules, i.e. operations, which are performed outside of the cube in Excel and the results of which are then written back into the cube.

Here is an example: Assume that we want to create two more elements in addition to those we created in our demo database earlier:

- Turnover EURO
- Turnover USD

You get the unit prices via the attribute "Price Per Unit". The turnover elements with two currencies should be filled using push rules.

Open the Modeller, and add two new elements to the dimension "Measures".
Create a view of the "Sales Cube" via "Paste View", which contains these new elements and a few of the product names.

Select "Measure". Select the elements as shown on below screen. Confirm the selection with "OK".
Proceed the same way as shown above for the product page. The result should then look as follows:

You already called the attribute cube "#_Products" once before in the "Information Processing with Attributes" section. Now we need this attribute as a parameter for the push rules. Retrieve the information in the first step.

Create a view of the attribute cube "#_Products" via "Paste View".
Select the element "_Price Per Unit". Confirm the selection with "OK".

For Row titles select the same products as previously. The following screen appears:

Now you get a display of the number of units and the elements (left) to be calculated. The product prices are shown in the table on the right.
You must now combine the information in one view. Use the view showing Units (left) and insert a blank row below row 2. Enter "#_Products" in there. Insert a blank column to the right of column A. Copy the attribute "_Price Per Unit" from the view on the right (B4) into the blank column "B" (cell B9).

The result appears as follows:

Enter the formula for the attribute "_Price Per Unit" in cell B10 utilizing the "Paste Data Function" from the Palo menu.

Note that you must place a checkmark in "Attribute cubes" in order to be able to select attributes. Select the attribute cube "#_Products". Also place a checkmark in "Guess Arguments".

This handbook has been personalized for NA - ohneanmeldung
Then the following display appears. Compare the formula, and you will see that Palo has "guessed" it correctly.

Copy the PALO.DATAC formula from cell B10 into cells B11 to B18.

The result appears as follows:

Please note: When you work with Push Rules, it makes sense to work with base elements.

This handbook has been personalized for NA - ohneanmeldung
So change the selection in cells A4 to A7 correspondingly and select "Germany", "Jan", "2006" und "Budget".

The display should then appear as follows:
When we think of currencies, we obviously need exchange rates. One way would be to enter these into the cube, but we want to make it easier at this point and enter them as fixed rates in Excel.

The columns D and E currently contain `PALO.DATAC` formulas. To write the results of our calculations into the database, we convert them into `PALO.SETDATA` functions.

The `PALO.DATAC` formula in D10 appears as follows, for example:

```
=PALO.DATAC($A$1,$A$2,$A10,$A$4,$A$5,$A$6,$A$7,D$9)
```

These parameters are the coordinates, which identify the cell in the database. We can use them in the same way for the `PALO.SETDATA` function. First simply replace the term `PALO.DATAC` with `PALO.SETDATA`.

The columns D and E currently contain **PALO.DATAC** formulas. To write the results of our calculations into the database, we convert them into **PALO.SETDATA** functions.
1st Step: PALO.DATAC becomes PALO.SETDATA:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>localhost/ Demo</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2</td>
<td>Sales</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A3</td>
<td># Products</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A4</td>
<td>Germany</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td>Jan</td>
<td>Exchange</td>
<td>Rate</td>
<td></td>
</tr>
<tr>
<td>A6</td>
<td>2006</td>
<td>USD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A7</td>
<td>Budget</td>
<td>1</td>
<td>1.254</td>
<td></td>
</tr>
<tr>
<td>A8</td>
<td>Price per</td>
<td>Units</td>
<td>Turnover</td>
<td>Turnover</td>
</tr>
<tr>
<td>A9</td>
<td></td>
<td></td>
<td>Euro</td>
<td>USD</td>
</tr>
<tr>
<td>A10</td>
<td>Desktop L</td>
<td>500</td>
<td>1,930.38</td>
<td>0.00</td>
</tr>
<tr>
<td>A11</td>
<td>Desktop Pro</td>
<td>600</td>
<td>5,580.64</td>
<td>0.00</td>
</tr>
<tr>
<td>A12</td>
<td>Desktop Pro XL</td>
<td>700</td>
<td>4,466.20</td>
<td>0.00</td>
</tr>
<tr>
<td>A13</td>
<td>Desktop High XL</td>
<td>750</td>
<td>2,460.03</td>
<td>0.00</td>
</tr>
<tr>
<td>A14</td>
<td>Desktop High XQ</td>
<td>800</td>
<td>2,965.48</td>
<td>0.00</td>
</tr>
<tr>
<td>A15</td>
<td>Server Power XC</td>
<td>850</td>
<td>4,496.39</td>
<td>0.00</td>
</tr>
<tr>
<td>A16</td>
<td>Notebook SX</td>
<td>1,500</td>
<td>7,500.00</td>
<td>0.00</td>
</tr>
<tr>
<td>A17</td>
<td>Notebook GT</td>
<td>1,800</td>
<td>14,541.70</td>
<td>0.00</td>
</tr>
<tr>
<td>A18</td>
<td>Notebook LXC</td>
<td>1,700</td>
<td>4,963.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

PALO.SETDATA requires two additional parameters which are to be added to the existing coordinates.

\[
\text{=PALO.SETDATA(B10\times C10\times D7, False, A1, A2, A10, A4, A5, A6, A7, D9)}
\]

- The parameter value, i.e. the value that is to be written to the database by PALO.SETDATA
- The second additional parameter is the Splash-parameter, which determines how we want to split (splash) if the formula refers to a consolidated element. An instruction, for example, could be: "Equally split the value among the subordinate elements. If we work with base elements, this flag is "FALSE", because nothing is to be distributed (broken down).

The value-parameter can be a number, a cell or a formula. In our case it is a formula. This parameter describes how to calculate the two turnover elements.

The entire variety of Excel functions is at your disposal. You can perform any type of calculation to set the content of a data cell. The Push Rule is not complex in our example. We simply multiply the number of units by the unit price and then multiply the sales turnover in Euros with the exchange rate.
2nd Step: PALO.SETDATA is completed by adding the two required parameters. 

3rd Step: PALO.SETDATA for currency conversion is also completed by adding the two required parameters

=PALO.SETDATA(D10*$E$7,FALSE,$A$1,$A$2,$A$10,$A$4,$A$5,$A$6,$A$7,D$9) 

Then copy the Palo.SETDATA formulas into the following cells D11 to E18. Please pay attention to the $-characters when making changes to the SETDATA formulas. These ensure that the formula can be copied correctly across the entire range.
After copying, take a closer look at the columns D and E. Here is the result:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Price per Unit</td>
<td>Units</td>
<td>Turnover Euro</td>
<td>Turnover USD</td>
</tr>
<tr>
<td>1</td>
<td>localhost/Demo_</td>
<td>Sales</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td># Products</td>
<td>Germany</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Jan</td>
<td>Exchange Rate</td>
<td>USD</td>
<td>1.264</td>
</tr>
<tr>
<td>6</td>
<td>2006 Budget</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Desktop L</td>
<td>500</td>
<td>1,330.38</td>
<td>965,190.00</td>
</tr>
<tr>
<td>11</td>
<td>Desktop Pro</td>
<td>600</td>
<td>5,550.04</td>
<td>3,330,904.00</td>
</tr>
<tr>
<td>12</td>
<td>Desktop Pro XL</td>
<td>700</td>
<td>4,446.23</td>
<td>3,112,396.00</td>
</tr>
<tr>
<td>13</td>
<td>Desktop High XL</td>
<td>750</td>
<td>2,450.03</td>
<td>1,066,622.50</td>
</tr>
<tr>
<td>14</td>
<td>Desktop High XQ</td>
<td>800</td>
<td>2,365.43</td>
<td>2,372,384.00</td>
</tr>
<tr>
<td>15</td>
<td>Server Power XQ</td>
<td>850</td>
<td>4,496.39</td>
<td>3,821,931.50</td>
</tr>
<tr>
<td>16</td>
<td>Notebook SX</td>
<td>1,500</td>
<td>7,500.06</td>
<td>11,250,090.00</td>
</tr>
<tr>
<td>17</td>
<td>Notebook GT</td>
<td>1,600</td>
<td>14,641.70</td>
<td>23,426,720.00</td>
</tr>
<tr>
<td>18</td>
<td>Notebook LXC</td>
<td>1,700</td>
<td>4,963.00</td>
<td>8,437,100.00</td>
</tr>
</tbody>
</table>

The price ("_Price Per Unit") has been multiplied by the number of units and the exchange rate. The result is in the database now. The returned value of the PALO.SETDATA function is identical to the applied value. It is displayed in the cell.

Remember that we described in the previous chapters that you can overwrite cells that contain PALO.DATA formulas. Enter a number in column B, and you see how the changes in "Price Per Unit" immediately affect the sales turnover columns.

**Note:** The calculated elements (Turnover EUR and Turnover USD) are based on the number of units, the "Price Per Unit" and the exchange rate. Unintentional changing of this parameter would result in immediate changes to the calculated elements. These dependencies can result in undesired effects.

Similar effects would be the outcome if there were no calculation and writing back of values into the database after a change. Consequently, you must ensure that the Push Rule is used consistently and throughout.

If you want to determine, for example, the effects of changes in the exchange rates, then do this in a separate worksheet. This makes sure that your current cube displays the correct results.

In the example just shown, we assumed a fixed exchange rate. But these rates are normally subject to constant changes. For example, these constant changes may result in automatic purchase and sales orders. A conventional example would be stocks: "Buy at a rate of x €." Of course, you can manage and check such automatic processes in your application. But you can also outsource these controls and calculations to a supervision server.

The Jedox supervision server monitors, controls and documents all data entries and changes in the Enterprise spreadsheets and within the Palo-Server. It can trigger warnings, alerts and correction operations and also initiate more far-reaching workflow or calculation processes. If you want to know more about the supervision server, visit us at http://www.jedox.com/en/products/palo_supervision_server/Introduction.html or write an e-mail to support@jedox.com.
5.10.4 Consolidation Factors

Whenever you worked with hierarchical structures, you already worked with consolidation factors even if you did not know it. The value in a consolidated element is calculated as the sum of the values in the subordinate elements. As long as this is done as a standard procedure, a factor (weighting) of 1 is assumed, each base value has the same weighting.

But you might want to weight the base elements differently to consider parts of multiples of base elements in a consolidated element. Negative weightings are also possible, e.g., when consolidating the base element Turnover and expenditures.

You can find an example in the "Sales" demo cube in the dimension "Measures". There is a consolidated element "Gross Profit", which is the difference between "Turnover" (sales) and "Cost of Sales". Click "Gross Profit" in the window on the left to display the elements to be consolidated.

You can also use the options which consolidation factors offer, for the following task: Imagine a report that is structured variably. Besides monthly results, you also want to be able to select quarterly or annual results. The report contains key figures that represent full-time employees. Of course, the number of employees is largely identical each month (permanent workforce). Consequently, the number of employees may not be added up in a quarterly report. Otherwise, a multiple of the workforce would be the result.

On the other hand, you have cost positions, for example. These figures need to accumulate to get a quarterly report (consolidation factor = 1). How can you take this requirement into consideration? Create two different consolidated elements per quarter (e.g., Quarter1 and Quarter1 cum.). You define an element with the factor "1" for each of the three months and total up the cost, for example. The second consolidated element has the same months. But each month is only allocated a share of 1/3 (factor = 0.333). As a result, you obtain the average number of full-time employees for all three months. Now you have the correct number in the sum total.

In the same way, you can now create the other three quarters and a special year element, which is composed of these four quarters with a consolidation factor of 0.25 each. This will allow you to perform all normal calculations using consolidation factors.
6 Entering, Changing and Deleting Data

6.1 Entering Data Manually

The easiest way to store data in Palo is traditional manual entry in an Excel-worksheet. Of course, this only makes sense for small data quantities.

You already made individual entries in previous tasks. An overview of the various options for data entry is presented below.

6.1.1 Entering in cells

Open a blank Excel-worksheet. Start "Palo/Paste View", and select the "Market" cube created earlier. Move "Months" into the Column titles and "Products" into the Row titles. Leave the other dimensions in the Page selector.

Double-click on "Select Elements…" of Months.
Then for Column titles select the first three months and click "OK".

Select the first five desktop computers from the “Products” dimension, and insert them as Row titles. You can find them under Stationary PCs.

The result appears as follows:

Here are displayed consolidated elements (Variance, Europe). However, for our first demonstration, base elements should be displayed instead.
To change the current settings, double-click Variance and change it to Actual. Change Europe to Germany in the same way.

The result should look as follows:

If you now enter a value, e.g., "175", it is written to the database.
Note that the formula is retained although you enter a value. The entered value is written to the database, and the formula is displayed again.

Entry of value "175"

The result of this entry:

As a result, you can enter data very easily to simulate scenarios, which are planned from bottom up. Of course, you also want to implement planning from top down. This is how you do it:
6.1.2 Entering Data in Consolidated Cells (Splashing)

Starting from our last example, double-click “Germany” and change it back to “Europe”. The result looks as follows:

First you see the values, which you entered before you changed the settings back to the consolidated view. The reason for this is obvious: the database was empty, you entered a number in the element “Germany” and consequently the sum for “Europe” is identical with the value for “Germany”.

Click another cell (e.g., C9) now, and try to enter something.

The following message appears:
**Splash Parameter #**

Ordinary writing into a consolidated cell is not possible. This is to prevent from writing data into a cell containing consolidated data unintentionally and thereby corrupt data. After all, consolidated data are the result of base element data.

However, to indicate that you are entering a value intentionally with the purpose to "splash", i.e. split the value among the subsequent base elements, start the entry with the special character "#", and then enter a value.

Enter the values to be splashed (#250 for Desktop L and #120 for Desktop Pro each in February):

```excel
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo_</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Market</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2002</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Europe</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Units</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Jan</td>
<td>Feb</td>
<td>Mar</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Desktop L</td>
<td></td>
<td>#250</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Desktop Pro</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>Desktop Pro XL</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>12</td>
<td>Desktop High XL</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>13</td>
<td>Desktop High XG</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
```

The values, which you entered:

```excel
<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo_</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Market</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2002</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Europe</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Units</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Jan</td>
<td>Feb</td>
<td>Mar</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Desktop L</td>
<td></td>
<td>175.00</td>
<td>#250</td>
</tr>
<tr>
<td>10</td>
<td>Desktop Pro</td>
<td>0.00</td>
<td>120.00</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>Desktop Pro XL</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>12</td>
<td>Desktop High XL</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>13</td>
<td>Desktop High XG</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
```

Values, which you entered, are split (broken down) and distributed across all levels below the current one. You can check that by scrolling through the regions.
The base values for "Germany":

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo Market</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2002</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Germany</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Units</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Enter the following data:
50 for "France" and "Desktop L" in January.

All other countries have value "0" in January.

Enter the following data:
50 for "France" and "Desktop L" in January.

All other countries have value "0" in January.

The resulting view for "Europe":

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo Market</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2002</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Europe</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Units</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If a value exists in at least one of the base elements, the total sum is split in line with the shares of the existing values. Assume that in addition to the value for "Germany" a value is entered for "France" in January, e.g., "50". All other countries in January equal "0".

When you enter a new value, this value is split among the base elements below that, in precisely the weighting of the previous values. If a value exists in one of the base related elements, then related base elements with the value 0 are not changed.
For example, if you change the total amount for Europe from 225 to 100 (44.4%) at "Desktop L", the value for Germany changes to 44.4% of 175, this is 77.78:

Respectively, it would be 22.22 instead of 50 for France. If no values existed previously, the new value is split equally among all subordinate base elements below the current level. This is shown for "Desktop Pro" in "Jan". Entry of the value 100 in the dimension "Europe" results here in an allocation of 5.26 to all base elements, for example to "France".

### Detailed view for "Germany":

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>localhost/Demo</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Market</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Germany</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Units</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jan</td>
<td>Feb</td>
<td>Mar</td>
<td></td>
</tr>
<tr>
<td>Desktop L</td>
<td>77.78</td>
<td>13.16</td>
<td>0.00</td>
</tr>
<tr>
<td>Desktop Pro</td>
<td>0.00</td>
<td>6.32</td>
<td>0.00</td>
</tr>
<tr>
<td>Desktop Pro XL</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Desktop High XL</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Desktop High XQ</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

### Detailed view for "France":

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>localhost/Demo</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Market</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2002</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>France</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Units</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jan</td>
<td>Feb</td>
<td>Mar</td>
<td></td>
</tr>
<tr>
<td>Desktop L</td>
<td>22.22</td>
<td>13.16</td>
<td>0.00</td>
</tr>
<tr>
<td>Desktop Pro</td>
<td>5.26</td>
<td>6.32</td>
<td>0.00</td>
</tr>
<tr>
<td>Desktop Pro XL</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Desktop High XL</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Desktop High XQ</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Besides "#", there are other parameters that you can use to write data in consolidated cells.

**Splash Parameter !**
An exclamation mark followed by a value: you *overwrite* all related base elements with this value. By entering "!10" in C10, the value 120 is replaced with 190(10*19 base elements).

**Entry of "!10":**

```
<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo Market</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Europe</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Units</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Actual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Jan</td>
<td>Feb</td>
<td>Mar</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>Desktop L</td>
<td>100.00</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>Desktop Pro</td>
<td>0.00</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>Desktop Pro XL</td>
<td>0.00</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>Desktop High XL</td>
<td>0.00</td>
</tr>
<tr>
<td>11</td>
<td>13</td>
<td>Desktop High XQ</td>
<td>0.00</td>
</tr>
</tbody>
</table>
```

19 * 10 = 190 replaces 120:
**Splash Parameter !!**
Two exclamation marks followed by a value add this value to the existing values of all related base elements. Consequently, entry of "!!20" would result in \(190 + 20 \times 19 = 570\) (adding 20 in 19 base elements).

Entry of "!!20":

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo Market</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2002</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Europe</td>
<td>Units</td>
<td>Actual</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Jan</td>
<td>Feb</td>
<td>Mar</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Desktop L</td>
<td>100.00</td>
<td>250.00</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>Desktop Pro</td>
<td>0.00</td>
<td>190</td>
<td>0.00</td>
</tr>
<tr>
<td>7</td>
<td>Desktop Pro XL</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>8</td>
<td>Desktop High XL</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>9</td>
<td>Desktop High XQ</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

\[190 + 19 \times 20 = 570\]
**Splash Parameter # … %**

Use this parameter to add or subtract percentage shares. #10% adds 10 % to the current value. By entering #10% in C10 we get 627 (570 +570*10%).

### Entry of "#10%":

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>570 + 570*10%</td>
<td>627</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

570 + 570*10% = 627:
Entering, Changing and Deleting Data

Since this chapter will later deal with data deletion and we need data to be deleted, apply above options again for practice purposes:

Change the year to 2003.
You can see that the months have zero amounts.

Enter the following values in January: 5000, 3000, 3500, 4000 and 6000.
The entry is in splashed format, e.g., #5000.

The result appears as follows:

Tip: Using these techniques, comprehensive changes are possible, it allows for simulations of the type, "What would happen if you change the data by x%?", for example. But this entails the danger that you may corrupt data: you no longer know the values of your original data and it cannot be retrieved again. However, there is a simple solution: create a backup of the database.
(See also chapter Backup of Databases).
6.1.3 Undo Data Input

As of Palo version 2.5, the Palo toolbar contains buttons, which allow you to undo data input. In order to make the Palo toolbar visible, click on a toolbar with the right mouse button, then activate the Palo toolbar.

For the undo-function, you find the following buttons in the toolbar:

- **Set reference for undo**
- **Undo one operation**
- **Undo all and cancel reference for undo**
- **Commit changes and cancel reference for undo**

Mark a cell-area, in which values from the cube are displayed, then push the button to set for this cells the reference for undo:

The other three buttons are now active.

The button "Commit" takes over the current values.

**Important:** After you set a reference for undo, it is not possible for other users to enter or change values in this area. Only after canceling reference for undo the area will be open for entries of other users. Before you quit Excel, you should also decide to commit or to undo all. Otherwise all changes are automatically undone when you quit Excel or when the user-session ends.
6.1.4 Copying Cell Contents (Copy & Like)

COPY & LIKE make it possible to copy values directly into another cell.

6.1.4.1 Copy

As an example, open "Paste View", and select in cube "Sales" the following:
Row titles = Years and within Year via "Select Elements…" select 2006, 2007 and 2008.
Column titles: Datatypes and within Datatypes via "Select Elements…" select Budget and Actual.

![Microsoft Excel - Book1](image)

Now you want to copy the Actual figures of 2007 to the Budget of 2008.
Go to cell B11 and write: Copy Actual,2007 or alternatively: Copy Datatypes:Actual,2007

![Microsoft Excel - Book1](image)
In both cases you get the following result:

![Microsoft Excel - Book1](image)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sales</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>All Products</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Europe</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Year</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Units</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2006</td>
<td>9,651,627.31</td>
</tr>
<tr>
<td>10</td>
<td>2007</td>
<td>10,616,790.04</td>
</tr>
<tr>
<td>11</td>
<td>2008</td>
<td>7,948,113.31</td>
</tr>
</tbody>
</table>

The sequence of the elements in the formula is irrelevant.

You may ask yourself if you could also enter the amount from Actual 2007 utilizing the sign "#" under Budget 2008 and what is the difference? The difference is that you transfer the source values including their splash structures 1:1 with COPY. With #, you captured and split this total value only.

But because Budget 2008 was holding a "0" previously, the split is equal among all base elements.

You can see the difference in the example of the element "Germany". As described earlier in the splashing section, the dimension "Europe" is composed of 19 base elements. Consequently, "Germany" will receive in Budget 2007 a share of 418,321.75 when split with the entry of "#7,948,113.31". On the other hand, it is 919,665.42 when using "Copy Actual, 2007".

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sales</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>All Products</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Germany</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Year</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Units</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2006</td>
<td>1,110,662.05</td>
</tr>
<tr>
<td>10</td>
<td>2007</td>
<td>1,221,728.26</td>
</tr>
<tr>
<td>11</td>
<td>2008</td>
<td>418,321.75</td>
</tr>
</tbody>
</table>

Uniform split with #

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sales</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>All Products</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Germany</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Year</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Units</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2006</td>
<td>1,110,662.05</td>
</tr>
<tr>
<td>10</td>
<td>2007</td>
<td>1,221,728.26</td>
</tr>
<tr>
<td>11</td>
<td>2008</td>
<td>919,665.42</td>
</tr>
</tbody>
</table>

Split according to shares with COPY
6.1.4.2 Like

The LIKE command contains the COPY command. It looks at the shares of the base elements when splitting the value placed before LIKE. Therefore, it allows for a different value formulation than COPY.

Example: The value for Actual at All Products/Europe/Year (2007) is 7,948,113.31. Consequently, 12000000 Like Actual,2007 splits the budget according to the shares in Actual 2007.

<table>
<thead>
<tr>
<th>Entry</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Microsoft Excel - Book1" /></td>
<td><img src="image2" alt="Microsoft Excel - Book2" /></td>
</tr>
</tbody>
</table>

Entry Result

If you shift the display from All Products/Europe/Year (2007) to Germany, you obtain the base values for Germany. These are, for example, 919,665.42 for Actual in 2007. Analogously, “Germany” has a budget 2009 of 12,000,000/7.948,113.31*919,665.42, this is 1,388,503.74.

To conclude, we want to show you that this method also works with consolidated elements. For example, “Gross Profit” is composed of Turnover (consolidation factor = 1) and Cost of Sales (consolidation factor = -1). Making an estimate on your Gross Profit will automatically bring along the planned estimates for the turnover and cost of sales accordingly.

This handbook has been personalized for NA - ohneanmeldung
You can see that the two base elements "Turnover" and "Cost of Sales" are filled accordingly.
6.2 Copying Data

Copying data via the clipboard in Pala is done via the `PALO.SETDATA` formula. You can find an exact description of the procedure in the "Importing from an Excel-worksheet" section in the "Importing and Exporting Data" chapter.

6.3 Deleting Data

If you right-click the cube in the Modeller, you can select between two different deletion options. You can delete the complete cube or defined data of the cube ("Clear cube" function).

Deleting the complete cube has already been described in the "Working with the Modeller" chapter. Therefore, we would like to restrict our explanation to clearing cubes at this point.

Now we want to delete these data. First you must set the corresponding parameters.
Call the Modeller. Click the "Market" cube, and then right-click "Clear cube".

Confirm and the following selection will appear:

Click the dots (…) next to the "Years" dimension.
Then you see a list of years. Select the year 2003.

This selection will be displayed in the "Clear cube" window.

Select "Clear selection". Make sure that the correct cube has been selected.
Confirm the question with "OK".

You receive a message as soon as the procedure is completed. The larger the cube, the longer clearing can take (a few minutes or longer).

Then you receive a message that the data were deleted (cube selection was cleared). Confirm by clicking "OK".

Then you return to the "Clear cube" dialog. You can clear other parts of the cube. Alternatively, end the procedure by clicking "Close".

When the “Clear cube” window appears, you can also clear the entire cube.

"Clear All" will clear the entire cube, the structure of the cube will remain intact. This will allow you to use the cube again and enter other values.

Thereafter, you must confirm the question with "OK".
You then receive a message confirming the procedure.

In the end, you may still see values in the Excel-worksheet. If that is the case, press function key "F9" to update the view.

Your worksheet should then look as follows:
7 Importing and Exporting Data

7.1 Importing Data

You created the cube "Market" in chapter "Working with the Modeller" ("Creating a New Cube"). This cube contains the same dimensions as "Sales". Consequently, the data structures of "Market" are a copy of "Sales". But "Market" does not yet contain data. This chapter explains how you can import external data into a new cube.

7.2 Importing using the Data Import Wizard

Principally, data can be imported in two different ways:

1. Import by utilizing the Data Import Wizard. It is helpful for specific standard sources.
2. Without the wizard. You can control and process the import in line with your requirements. In addition, special types of transfer options will be described.

7.2.1 Importing Text Files

Text files provide one of the simplest ways to read data via importing.

Please bear the following important note in mind: In this description, we first import data cells and then dimensions. Of course, this only works, because we can access demo data and an already created cubes. On the other hand, if you were to create a model, you would use the following sequence:

1. Prepare by planning the structures
2. Create a new database
3. Create dimensions
4. Create a cube with the dimensions
5. Importing elements into the dimensions
6. Import data cells

7.2.1.1 Importing Data Cells

You get a file called "Testimport.txt" together with the Palo setup. You can find the file in the folder C:\Documents and Settings\All Users\(Shared )Documents\Jedox\Palo. The file contains approx. 31,000 data records, which appear as follows:

<table>
<thead>
<tr>
<th>Switzerland</th>
<th>Keyboard GT</th>
<th>2003</th>
<th>Apr</th>
<th>Actual</th>
<th>572</th>
</tr>
</thead>
<tbody>
<tr>
<td>Switzerland</td>
<td>Keyboard GT</td>
<td>2003</td>
<td>Mai</td>
<td>Actual</td>
<td>572</td>
</tr>
<tr>
<td>Switzerland</td>
<td>Keyboard GT</td>
<td>2003</td>
<td>Jun</td>
<td>Actual</td>
<td>572</td>
</tr>
<tr>
<td>Switzerland</td>
<td>Keyboard GT</td>
<td>2003</td>
<td>Jul</td>
<td>Actual</td>
<td>336</td>
</tr>
<tr>
<td>Switzerland</td>
<td>Keyboard GT</td>
<td>2003</td>
<td>Aug</td>
<td>Actual</td>
<td>572</td>
</tr>
<tr>
<td>Switzerland</td>
<td>Keyboard GT</td>
<td>2003</td>
<td>Sep</td>
<td>Actual</td>
<td>1008</td>
</tr>
<tr>
<td>Switzerland</td>
<td>Keyboard GT</td>
<td>2003</td>
<td>Nov</td>
<td>Actual</td>
<td>1008</td>
</tr>
<tr>
<td>Switzerland</td>
<td>Keyboard GT</td>
<td>2003</td>
<td>Dez</td>
<td>Actual</td>
<td>1000</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>Notebook SX</td>
<td>2002</td>
<td>Jan</td>
<td>Actual</td>
<td>14175</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>Notebook SX</td>
<td>2002</td>
<td>Mrz</td>
<td>Actual</td>
<td>9450</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>Notebook SX</td>
<td>2002</td>
<td>Apr</td>
<td>Actual</td>
<td>18900</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>Notebook SX</td>
<td>2002</td>
<td>Mai</td>
<td>Actual</td>
<td>4725</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>Notebook SX</td>
<td>2002</td>
<td>Jul</td>
<td>Actual</td>
<td>14175</td>
</tr>
</tbody>
</table>

This is the result of a typical data export from a database, which contains targeted and actual sales figures from various countries and years. The content is comparable with that of the cube "Sales", the structures of which you can call via the Modeller.

Now we will show you how you can use these data for data import. In this case, we work with the demo database and the cube "Market" created therein. Therefore, you can start the Import Wizard immediately.

This handbook has been personalized for NA - ohneanmeldung
Open a blank Excel-worksheet, and click "Data Import..." in the Palo menu. Then the following window appears:

You have four options for the data import:

- Flat Textfile: imports a file in txt or csv format. The data are separated by semicolon or a tab.
- ODBC Query: import data from an ODBC data source.
- Cube: data are imported from another cube.
- Internal Loop: a list on a worksheet is processed via a loop.

Let's start with the text file. Click "Next".
You are asked for the data source. Click "Browse" to select the "Testimport.txt" file. It can usually be found in the following folder: C:\Documents and Settings\All Users\(Shared )Documents\Jedox\Palo.

Enter the separator character: the columns are separated by tabs in this example. There are no decimal points or header lines in the test data, consequently, the corresponding options can remain blank.

Click "Next" to access the following screen.

Click "Next" again and then "Close". Then you see the following screen in the Excel-worksheet:

This is the first data record of our text file.
If you clicked the button “Finish” instead of “Next” or “Close”, all data records would be written into the first row of our Excel-worksheet. The worksheet would be recalculate after each data record, and the process would be repeated until the last data record.

But we do not want to have the data in an Excel-worksheet but transfer the data to Palo instead. Consequently, we must create a formula that will transfer the contents of the first data record or first Excel row into the Palo cube.

You will remember that we showed in chapter “Creating a Palo Worksheet” in section “Paste Function” how Palo can “guess” the arguments of a function. Now we want to use this option.

Click in a row below the first row, in D4, for example. It is important that the cell is in a column between the first and last field of the to be imported data record. Now click “Paste Data Function”. Select the database, the cube and “Guess Arguments”.

When you then press F2, you see the formula.
Most arguments in the formula have been identified correctly: We must correct the remainder manually:
The cell F1 contains the value to be written.
With "TRUE" or "FALSE" you can decide, whether it is permitted or not to write the value into a
consolidated cell. With "FALSE", it can only be written into base elements.

In addition, "Variance" is not applicable. "Variance" is a consolidated element, which gives the difference
between "Budget" and "Actual". Therefore, E1 contains a different name for "Budget". We must correct
this:
You might wonder why there is an element with the name "Planned" in the import file.
We have set it up this way to show you how the DATAC function works when there are discrepancies
between old and new names. Such discrepancies can occur, for example, when you import data from
other data sources where other names are used than in Palo cubes. But you do not want to import the
other name "Planned" into the cube, but instead use your name "Budget".

To achieve this, define an import rule: If E1 = "Planned", then select the element "Budget", otherwise =
E1. Write the following formula in E2:

=IF(E1="Planned","Budget",E1)

By this, we instruct the PALO.SETDATA-function to use "Budget" instead of "Planned".

Then change the DATAC function in F4 by replacing the incorrect entry "Variance" with a reference to E2:

Then select the "Data Import Wizard" again, and test whether the formula is processed correctly for the
remaining data records. Your previous settings still exist. Double-click "Next". You see the import window
again. Then click "Next" a few times and observe how the formula changes in cell F4.
After a few data records, you get this message:

![Microsoft Excel screenshot](image)

Again, the PALO.SETDATAC-function does not retrieve a cube cell with the coordinates it has. Close the wizard, and call the context menu.

A right-click in cell D4 will display the following menu.

![Excel context menu](image)

Select "Palo Error" to display the error message:

![Palo Error dialog box](image)

This handbook has been personalized for NA - ohneanmeldung
Examine the elements in the dimension "Months" and you will see that the English term "Mar" is used instead of the German abbreviation "Mrz" in the database. Consequently, we must provide a translation. See the screen below:

Now change the PALO.SETDATA- function in cell D4 again. Replace $D$1 with $D$2 to use the English abbreviations of the months.

Then select the Data Import Wizard again, and check whether the formula is being processed correctly.

**Tip:** In a few installations the value in F1 is not recognized as a number. This also applies if the field is formatted correctly otherwise. In this case, Palo outputs an error message (e.g., "Cannot write text in a numeric cell"). To get around this, enter the formula "=Value(F1)" in cell G1, and change the "PALO.SETDATA" function as follows:

```
=PALO.SETDATA($G$1,FALSE,"localhost/Demo","Market",$D$2,$C$1,$A$1,$E$2,"Units",$B$1)
```

Save the worksheet and start Data Import again. To be sure everything is correct, step through a number of record imports by clicking "Next". If the write is performed properly, PALO.SETDATA() will return the value which it has written into the cube.

By using the import wizard data can be auto-formatted during the import process. This happens by using the button "Next". The format of a cell is applied when pressing "Next" while it is ignored with "Finish". This auto-formatting of cell A1 changes the imported values to the regional settings (e.g. date, currency, number format) that furthermore can lead to problems in depending formulas. To avoid the formatting feature set the cell format to the format of the source or to "Text".
The data will be properly written to the cube, provided, the value in cell D4 (PALO.SETDATA"-function) always equals the value in cell F1 and cell G1, respectively.

Once you are sure that no further problems are to be expected, click “Finish” in the "Data Import Wizard". If you remove the checkmark next to "Refreshing screen while finishing", the import process is faster.

---

This handbook has been personalized for NA - ohneanmeldung
**Important:** In case you need to know if an import of data is currently in progress, you can use the variable `_paloimportactive`. You can find it in the Excel menu under Insert/Name/Paste as soon as the first import has started.

![Variable _paloimportactive](image)

This information can protect you from writing data with the PALO.SETDATA-function inadvertently into the cube. Every time, if you click into the Formula Bar and confirm this function with Enter or if you press F9, this function is executed.

**Tip:** Therefore, use the following formula in D4:

```excel
=if(_paloimportactive,PALO.SETDATA($G$1,FALSE,"localhost/Demo","Market";$D$2;$C$1;$A$1;$E$2,"Units";$B$1))
```

In this form the PALO.SETDATA-function is only executed when the Data Import Wizard runs.

We insert this name into the cell G4. You get the value “TRUE” during import.
Once the import has been completed, the display changes to "FALSE". This information seems to be meaningless at first glance. After all, the progress bar will show you when an import is being processed. But indication of the status can also be required by other Palo functions, i.e. they only process if the status of `_paloimportactive` is equal to the value intended for that particular function.

You can check whether the last data record has been input correctly. Therefore open a blank Excel-worksheet and create the following overview with "Paste View":

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>United Kingdom</td>
<td>Keyboard GT</td>
<td>2003</td>
<td>Dez</td>
<td>Actual</td>
<td>3024</td>
<td>3024</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>Dec</td>
<td></td>
<td>Actual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3024</td>
<td>FALSE</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For comparison: the last data record after the import has been completed.

Therefor open a blank Excel-worksheet and create the following overview with "Paste View":

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Market</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Dec</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2003</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>United Kingdom</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Units</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Actual</td>
<td>3024.00</td>
</tr>
<tr>
<td>9</td>
<td>Keyboard GT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 7.2.1.2 Update import

As a further examples, we assume, that you should update the "Market-data". For it, you read the current values from the cube with `PALO.DATA()`, then you add the values from the file "updateimport.txt" to them (path: C:\Documents and Settings\ All Users\ Shared Documents\Jedox\Palo\updateimport.txt). G4 contains "-paloimportactice" and the function in D4 is:

```
=IF(G4, PALO.SETDATA(PALO.DATA("localhost/Demo","Market",D2,C1,A1,E2,"Units",B1)+G1,FALSE,"localhost/Demo","Market",D2,C1,A1,E2,"Units",B1))
```

This handbook has been personalized for NA - ohneanmeldung
The following screenshot shows the last record of the file "updateimport.txt". B4 contains the current value of the corresponding cube cell: =3024 + 3000:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 United</td>
<td>Kingdom</td>
<td>Keyboard GT</td>
<td>2003</td>
<td>Dec</td>
<td>Actual</td>
<td>3000</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>Dec</td>
<td>Actual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>6024!</td>
<td>FALSE</td>
<td></td>
</tr>
</tbody>
</table>

**7.2.1.3 Importing Comments**

With PALO.SETDATA() you can import comments just as you can import data values. Note, however, that the PALO-functions, for reasons of compatibility with MS Excel 2000 and former versions, keep a representation limit of 255 characters per Excel-cell. In chapter 5.9.2 (Display of Long Comments), it was already explained, how you nevertheless can write longer comments into cube cells and how you can show them in the Excel-worksheet by using PALO.DATAT() as an array-function. Because PALO-SETDATA() is limited to 255 characters, you'll get the error message "#value!" in case you try to import comments which are longer than 255 characters. In this case, nothing will be written into the cube cell.

**7.2.1.4 Importing Elements**

The Palo function PALO.EADD enables the import of dimension elements from a text file. We have again prepared an example to demonstrate how the import works. In this example, we import the elements in one step to the three dimensions: "Time", "Costcenter" and "Region". To avoid problems with other demo data, create a new database, which contains these three dimensions.
Click on "Modeller..." in the Palo-menu – there you can start the Palo Wizard. With the Palo Wizard, you can create a new database:

After you click "Next", enter "Demo1" for your database name.
Click "Finish" and you get a confirmation that database was created successfully.

Now you need to fill it.

Open the Modeller. Select the new database, and create the three dimensions "Time", "Costcenter" and "Region":

In the Modeller, also create a new cube and call it "Demo cube", for example. This cube contains the dimension you just created. Close the Modeller.
You can find an Excel file with the name metadataimport.xls in the folder C:\Documents and Settings\All Users\(Shared)\Documents\Jedox\Palo. Open this file:

![Excel spreadsheet](image)

You can see that row 1 is blank. You saw in the previous section "Importing Data Cells" how data are imported. The Palo data import function populates this row line by line with the contents of the text file "metadata.txt" (in the same folder). The data records contained look as follows:

![Spreadsheet data records](image)

You will find the headers again in row 2 of the spreadsheet. The headers are mainly shown to make it more transparent what is happening in the worksheet. Only cell A1 is referenced in a formula (in cell B5). If A1 is identical to A2, it is assumed that the currently imported row is a header row and should be skipped.

This handbook has been personalized for NA - ohneanmeldung
Cell B4 contains the name of the database created earlier. Make sure that the name is localhost/Demo1. To understand what the formulas do in this worksheet, start importing the first data record. Click the "Data Import Wizard" to open it. Then select Flat Textfile (*.txt, *csv) and click "Next".

You can find the "metadata.txt" file in the folder C:\Documents and Settings\All Users\(Shared\)Documents\Jedox\Palo. The fields of this file are separated by tabs. Make sure that the option "Tab" is selected, and click "Next". When you click "Next" again, the first row of the import is read. It contains the headers:
Click "Next" again to read the first real data record.
Now let's take a closer look at the three areas in which the elements are calculated and written into the cube. Let's start with the fields under "Time Dimension". There you see Month, Quarter and Year, which are results of the data field in cell A1.

To read data in line with OLAP terms, you must pay attention to the following. The base element is the month. The months are consolidated to quarters. These are in turn consolidated to years. Now let's take a look at the PALO.EADD function in cell B13. It returns "FALSE" as long as the import loop is not executed or if the import is aborted or complete.

The "If query" in the formula ensures that the header row is skipped.

=IF(NOT($B$5),PALO.EADD($B$4,"Time","N",B9,B10,1,TRUE))
PALO.EADD() requires the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server / database</td>
<td>Text</td>
</tr>
<tr>
<td>Dimension</td>
<td>Text</td>
</tr>
<tr>
<td>Type</td>
<td>You can always use type &quot;n&quot; (base elements). The element type will be converted to ‘c’ anyway in case the import loop assigns sub elements to it later on.</td>
</tr>
<tr>
<td>Element name</td>
<td>Text</td>
</tr>
<tr>
<td>Parent name</td>
<td>Text</td>
</tr>
<tr>
<td>Weight</td>
<td>Number (describes the consolidation factor)</td>
</tr>
<tr>
<td>Clear</td>
<td>Possible parameters values are 0 (or FALSE), 1 (or TRUE) and 2 0: Removes before importing no elements of the dimension. 1: Removes before importing all elements of the existing dimension. 2: Removes before importing all existing elements of the C-dimension, basic elements are not deleted. <strong>Note:</strong> Make sure that all EADD functions referring to the same dimension use identical parameters (true or false). Otherwise, Palo would execute that parameter for the complete dimension that happens to be found first. Reason: the sequence, in which Excel formulas are processed, cannot be determined. During an import the command delete will only be executed one time with the first PALO.EADD() and then it will be ignored.</td>
</tr>
</tbody>
</table>

The formula in cell B13 states with this: "Jan" is an element of the dimension "Time". Its parent element (consolidated element) is "Qtr. 1", and the consolidation factor in Qtr. 1 is 1.

The PALO.EADD formula first executes the query for TRUE. TRUE means that any existing elements of the dimension "Time" are deleted (e.g. old entries). Then the two elements "Jan" and "Qtr" are created.

If PALO.EADD() has to create an element, which was already created by another cell entry with PALO.EADD() in the same import, then in creation elements happens nothing. Only that "N" will be changed to "C" if it is necessary.

Now you also understand why the formula beneath in cell B14 is:

```
12 | 13 | Formula 1 | FALSE |
14 | Formula 2 | =IF(NOT($B$5),PALO.EADD($B$4,"Time","n",B10,B11,1,TRUE)) |
15 |
```

"Qtr. 1" is an element of the dimension "Time". Its parent (or consolidated) element is "2005", and the consolidation factor in 2005 is 1. You can find a description of consolidation factors in the "Consolidation Factors" section of the "Working with the Modeller" chapter. You can find another example of working with consolidation factors in the "Like" section of the "Entering, Changing and Deleting Data" chapter.

Both formulas together describe the allocation of months to quarters (in B13) and of quarters to years (in B14). And therefore the following applies: while one PALO.EADD formula describes two elements and their relationship to each other, two formulas are required to describe three elements of a dimension.
The procedure in the next dimension "Costcenter" is similar. The text file contains costcenter numbers, which are base elements. The next level is the Costcenter Group. Its description is stored in the cells D18:D26 of the table. In addition, there is a "Total" level as the highest level.

We again have two formulas that describe three elements, two of them at the same time each time.

Import of the third dimension "Region" functions somewhat differently since it is asymmetric.

The dimension "Region" contains three types: Countries, Subregions and Regions. You will see that the European countries are consolidated to the subregions South, East and West before these Subregions are again consolidated into Europe. But the countries of the regions America and Asia are consolidated to the respective region without a previous allocation to the "Subregions" level.

The result in Palo after the import of all data appears as follows (a few subregions are displayed in abbreviated form for legibility reasons):
Now check how the formulas are set up:

The formula in cell B34 is:

\[
\text{IF( NOT($B5), IF(B31<"","", \text{PALO.EADD($B4,"Region","N",B30,B31,1,TRUE)}))}
\]

It excludes the processing of header rows due to the first if-query. The second if-query ensures that this formula is only executed if a subregion exists. It defines a subregion as parent or consolidated element of Country. Otherwise, the formula in B35 combines the subregion(s) (if existent) or the countries into a region.

Now start data import, and click "Next" a few times. After you determine that everything runs as intended, click "Finish" to conclude the structure import.

The next step would be the import of the data in column "G". Please refer to the previous section on the import of data cells.

**Note:** Do not use \text{PALO.SETDATA} and \text{PALO.EADD} together in one worksheet. That would result in the same issue as already addressed in the description of the parameters for the \text{EADD} function. The sequence, in which formulas are executed, cannot be determined beforehand. It may read values, for which no dimensions have been created.

**Note:** It only makes sense to use the function \text{PALO.EADD} in connection with an import, because the import wizard recalculates the database.

**Tip:** You know that you should not use \text{PALO.SETDATA} and \text{PALO.EADD} together in one worksheet. It is possible to use these functions in different worksheets of one workbook. But the \text{EADD} functions must then end as a mandatory condition at "FALSE". Consequently, this procedure is not recommended.

It is better if you process both functions in separate Excel-workbooks. This also has a practical benefit, because you will probably need to import with \text{SETDATA} functions more often in the future, especially if you import the same type of data from other sources at regular intervals. On the other hand, you only need the \text{EADD} functions for the initial setup of elements or for the import of new elements. If you proceed in this way, you can execute the import of elements using \text{EADD} functions. Then close this workbook (e.g., Import Element.xls) and open the workbook with the \text{SETDATA} functions (e.g., Import Data.xls) and perform the corresponding imports. This prevents conflicts between the formulas, which may otherwise result due to the interdependencies.
7.2.1.5 Deleting Elements

You can use the function `PALO.EDELETE` to delete elements. It uses the same options as described in connection with the import of text files.

From our previous examples, we still have "Customer1" … "Customer6" in the dimension "Customers" in our demo-database. Let's assume that the customers "Customer2" and "Customer5" have bought nothing from us in the last two years. Therefore we would like to remove them from the database. We can do that with the PALO.EDELETE-function:

Proceed as follows:


Enter "yes" in the cells B2 and B5.

Now save the worksheet as a text file. Name the file, e.g., "Delete elements .txt". Close the saved text file.

Open another blank Excel-workbook. Enter the `PALO.EDELETE` function in cell C3 as follows:

```
=If(B1="Yes",PALO.EDELETE("localhost/Demo","customers",$A$1),"")
```

Deletion is only carried out if "yes" is in the import cell "B1". In this case, the element with the name of the customer in cell "A1" is deleted.

Your worksheet then looks as follows:

![Excel spreadsheet with PALO.EDELETE function applied](image)

Now import the text file. Proceed as described in the "Importing Text Files" section.

1. Open the "Data Import Wizard"
2. Select "txt" as file format.
4. Leave the other settings, and click "Next".
The import is executed row by row. If "Customer2" is shown, the formula returns "True".

If the import is successful, click “Finish”.

Now call the Modeller.

Now you see, that the customer "Customer2" and the customer "Customer5" have been deleted, as desired.
7.2.1.6 Renaming Elements

You just got to know the **PALO.EDELETE** function. If you want to rename larger quantities of elements, the renaming by means of the Modeller is too time-consuming. For this purpose, you have **PALO.ERENAME**. This is another function that should only be used with the Import Wizard.

Take the same dimension as with the deletion of elements. Now, "Customer1" should be renamed in "Miller" and "Customer6" in "Smith".

Now use the same procedure as for deleting elements:

Open a blank Excel-worksheet, and enter the names "Customer1" and "Miller", "Customer6" and "Smith". The worksheet should look as follows:

Now save the worksheet as a text file.
Name the file, e.g., "Rename elements .txt".
Close the saved text file.

Open another blank Excel-worksheet. Enter the following function in cell B3:

```
=PALO.ERENAME("localhost/Demo","Customers",A1,B1).
```

The Excel-worksheet looks as follows:

The "FALSE" in cell B3 means that the import isn’t active.

Now import the text file. Proceed as described in section "Importing Text Files".
- Open the "Data Import Wizard"
- Select "txt" as file format.
- Search for the "Rename elements .txt" file.

Leave the other settings as they are, and click "Next".
The import is executed row by row:

If the import is successful, click "Finish".

Call the Modeller:

You will see that the renaming of element "Customer1" into "Miller" and "Customer6" into "Smith" was successful.

7.2.2 Importing from ODBC Sources

The import technique that you used for text files, can also be used in a similar form for other data sources based on ODBC. Together with Palo you get a Microsoft Access database. It contains the same data as the "Testimport.txt" file. It is called "Testimport.mdb" and is located in the folder C:\Documents and Settings\All Users\Shared Documents\Jedox\Palo by default.
7.2.2.1 Link to ODBC Source

Before you can import data from an ODBC database, you must log on to the database source name (DSN). You can do this quickly in Windows XP; only four steps are required.

- Select Start/Control Panel/Administrative Tools/Data Sources (ODBC).
- Click “System DSN” and select “Add”.
- Select the Microsoft Access driver, and click “Finish”.
- Enter “Palotest” as data source name in the next menu. Enter “Testimport.mdb” in the Description field and also select this file as your database.

7.2.2 Importing in Palo

After you have done the settings, open a blank Excel worksheet.
Select the cell A1, and click “Data Import Wizard” in the Palo menu.
Select “ODBC Query” as data source.
You see the following screen after you click "Next". The DSN field may still read "MS Access Database", instead select "Palotest".

Because no user name or password has been assigned, these fields may remain blank. To retrieve data from the database, an SQL query must be formulated in the text box. Enter the following SQL statement under 2: SELECT * FROM Sheet1.

This entry retrieves data from a table called "Sheet1". "Sheet1" is the only table in the Access database. It contains all data that we want to import.

The result appears as follows:

Note that you have the option to store all SQL statements. This option allows you to reuse them later and you will not need to recreate them repeatedly.
After you click "Next", you see the import step which you already know from importing text files.

The following procedure is the same as described in section "Import of Textfiles".

### 7.2.3 Importing from Other Cubes

We created the cube "Market" when working with the Modeller and already used it for a few tests. At the same time, we confirmed that the cube "Market" is a copy of "Sales". In such a case, you can export data from a cube and then import into another one without problems. But by the same token this also means that this type of import is only possible if the structures of the cubes are identical.

The cube "Market" already contains data for the years 2002 and 2003 after the earlier import of the text files. Consequently, we now want to create a view, which shows the year 2004. Create a blank Excel-worksheet, and chose a few elements from "Products" in the Row titles and "Units" (dimension "Measures") in Column titles. Then set your view to "Actual", "Germany" and "2004". The result appears as follows:
Now switch to a blank table in the current Excel-workbook. Call the "Data Import Wizard", and set "Cube" as source.

You see the following screen after you click "Next".

Your personal work environment determines the appearance of your screen. But you should be able to select "Demo" as your database and "Sales" as your cube in the "Choose Server/Database" and "Choose Cube" dropdowns.

This handbook has been personalized for NA - ohneanmeldung
When you select these parameters, the screen display changes as follows:

Additionally select "Germany" for "Regions" and "2004" for "Years".

In the next window, you could change settings for the import. We leave it to the default:
When you click "Next", you get to the Data Import Wizard which will read the individual data records.

Clicking "Next" writes again the first data record of the import cube into your table (starting with A1).

At this point we need the PALO.SETDATA function. Click "Close".
Enter a PALO.DATAC formula in cell G3 using "Palo - Paste Data Function...". Make sure that you select the cube "Market" and that "Guess Arguments" is activated.

Correct the formula as follows:

=PALO.SETDATA(G1,FALSE,"localhost/Demo","Market",$C$1,$D$1,$B$1,$E$1,$F$1,$A$1)
Restart the "Data Import Wizard". You might have to select the previously made settings again. When you get to this import step again, click "Next" a few times. Check whether the import is processed properly. If yes, click "Finish". Once the import procedure is completed, switch back to the table that you created earlier. You will now find the values for the units.

If you want to verify whether this data is the same as contained in the "Sales" cube, just copy the area A1 to B12, and paste it to a section below, e.g. starting in cell A14. Replace the cube name "Market" in A15 with "Sales". Select the area B22:B25 and delete the contents of these cells.

The screen shown here to the right appears:
Click cell B22, and call "Paste Data Function".

Select "Sales" as your cube this time. Do not forget to select "Guess Arguments", and then click "Paste". The following screen appears:

When you click the Formula Bar of Excel, you can see the contents of the PALO.DATAC function. Palo now accesses the "Sales" cube.

If you compare the functions in cells B9 and B22, you will find that with the exception of the cube name and the order of arguments (arises from the different dimension sequences in the cube) are identical.

This handbook has been personalized for NA - ohneanmeldung
The formula for the cube "Market" is:

=PALO.DATAC($A$1,$A$2,$A$3,$A$4,$A$5,$A$6,B$8,$A9)

or

=PALO.DATAC("localhost/Demo","Market","Year","2004","Germany","Actual","Units","Desktop L")

The formula for the cube "Sales" is:

=PALO.DATAC($A$14,$A$15,$A$22,$A$5,$A$3,$A$4,$A$6,B$21)

or

=PALO.DATAC("localhost/Demo","Sales","Desktop L","Germany","Year","2004","Actual","Units")

As a result, data from the cube "Sales" have been imported successfully into the cube "Market". To conclude, copy the formula from cell B22 to B23:B25. You now see the same values for all products in both cubes.
7.2.4 Internal Loops

With Internal Loop a counter loops in cell A1 and the worksheet executes functions until an error occurs. You can import data lists easily in this manner.

7.2.4.1 Preparatory Work

We start by adding the region "Russia" in the Modeller. Do not worry about consolidating it to "Europe", right now we are only interested in demonstrating the functionality.

Open the "testimport.txt" file using Excel. This in the following folder as a default: C:\Documents and Settings\All Users\(Shared)\Documents\Jedox\Palo.

You have to make a few changes in the document. First, save the file under the name "Russia.xls". Proceed as follows:

1. Select column A, and insert a blank column.
2. Select rows 1 and 2, and insert blank rows.
3. Starting at cell B59 delete all following rows.
4. Change the text from "Austria" to "Russia" in column B. Select the area B3:B58, and then press CTRL+H. Enter the text, and click "Replace all".
5. Change the text from "Planned" to "Budget" in column F. Select the area F3:F58, and then press CTRL+H. Enter the text, and click "Replace all".
6. Change the month names to English spelling in column E. Using the method just described, replace "Mrz, Mai, Okt and Dez" with "Mar, May, Oct and Dec" in E3:E58.

Save the file again. You should now have a file with a total of 56 data records starting with cell B3. Consequently, the loop must be ended after the 56th data record has been processed or when the 57th data record is read.
The "Russia.xls" file now appears as follows (excerpt):

To check the import later, create a blank Excel-worksheet.
You create the following view with "Paste View":

Drag the "Products" into the Row titles, and select the notebooks displayed above.
Transfer "Months" into the Column titles, and select the individual months. After the pasting, please also switch "Europe" to "Russia" and "Variance" to "Budget". This makes it possible for you to check how the loop is processed.
Then go to cell B1 and insert the following function:

\[ \text{If}(\text{OFFSET}(\$B\$3,\$A\$1-1,0)<>'',''Please wait...''),10/0) \]

Insert in C1:
\[ \text{OFFSET}(\$B\$3,\$A\$1-1,0) \]

Insert in D1:
\[ \text{OFFSET}(\$B\$3,\$A\$1-1,1) \]

Insert in E1:
\[ \text{OFFSET}(\$B\$3,\$A\$1-1,2) \]

Insert in F1:
\[ \text{OFFSET}(\$B\$3,\$A\$1-1,3) \]

Insert in G1:
\[ \text{OFFSET}(\$B\$3,\$A\$1-1,4) \]

Insert in H1:
\[ \text{OFFSET}(\$B\$3,\$A\$1-1,5) \]

We use the "OFFSET" function in the cells B1 to H1 to return the reference values of a defined area. Starting at cell B3 as the reference, the Excel function "OFFSET" will choose the row which is the result of the start reference + the current value of A1-1. Because A1 is still currently blank, the result = "DIV/0". The value behind "$A$1-1" states whether the start cell is in cell B3 or a column moved by "X".

The formula in cell B1 is:
\[ \text{If}(\text{OFFSET}(\$B\$3,\$A\$1-1,0)<>'',''Please wait...''),10/0) \]

The if-query checks whether the value in B3 is blank (<>''). As long as B3 is not empty, B1 displays "Please wait...". Otherwise an error is evoked in B1 because 10 divided by 0 results in "#DIV/0" and with this error the process is aborted.

The formula in cells C1 to H1 states (using the example of D1):
\[ \text{OFFSET}(\$B\$3,\$A\$1-1,1) \]

This formula is identical for all cells. The only difference is in the last part of the formula. Here you define whether the reference value should be taken from a column moved by "X". Consequently, the formula states:

Start reference = $B$3
Counter = $A$1-1 = move the start reference by the number resulting from A1-1
Return: the column (+1), which follows the start reference + counter, i.e. column "C" in this case.
Colour-coding is only to make the table easier to read. The cell "I1" is to contain the formula that writes your data into the "Sales" cube. You are already familiar with the procedure.

Insert a **PALO.SETDATA** formula in cell "I1" using "Paste Data Function".

The following formula results (cf. Excel Formula Bar):

<table>
<thead>
<tr>
<th></th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>#DIV/0!</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>Russia</td>
<td>Notebook SX</td>
<td>2002 Jan</td>
<td>Budget</td>
<td>6300</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Russia</td>
<td>Notebook SX</td>
<td>2002 Feb</td>
<td>Budget</td>
<td>12600</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Russia</td>
<td>Notebook SX</td>
<td>2002 Mar</td>
<td>Budget</td>
<td>6300</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Russia</td>
<td>Notebook SX</td>
<td>2002 Apr</td>
<td>Budget</td>
<td>12600</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Russia</td>
<td>Notebook SX</td>
<td>2002 May</td>
<td>Budget</td>
<td>6300</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Russia</td>
<td>Notebook SX</td>
<td>2002 Jun</td>
<td>Budget</td>
<td>9450</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Replace in:

=PALO.SETDATA(value,FALSE,"localhost/Demo","Sales","All Products","Europe","Year","2002","Variance","Units")

value with H1
"All Products" with D1
"Europe" with C1
"Year" with F1
"2002" with E1
"Variance" with G1

Your PALO.DATAC formula then appears as follows:

=PALO.SETDATA(H1,FALSE,"localhost/Demo","Sales",D1,C1,F1,E1,G1,"Units")

As long as cell A1 is still blank, the formulas in the area B1:I1 return errors messages or "0". If you enter "1" in A1, you obtain the following:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Please wait</td>
<td>Russia</td>
<td>Notebook SX</td>
<td>2002 Jan</td>
<td>Budget</td>
<td>6300</td>
<td>6300</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Russia</td>
<td>Notebook SX</td>
<td>2002 Jan</td>
<td>Budget</td>
<td></td>
<td>6300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Russia</td>
<td>Notebook SX</td>
<td>2002 Feb</td>
<td>Budget</td>
<td></td>
<td>12600</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If the "OFFSET" function is executed in C1 to H1, you obtain the values of B3 to G3. If you enter "2" in A1 instead, you obtain the values B4 to G4.
7.2.4.2 Importing in Palo

Of course, it would not be very convenient if you now had to enter all values from 1 to 56 manually into cell A1. Delete the value entered in cell A1, and start the Data Import Wizard. Then select “Internal Loop”.

If you click twice on "Next", you will get:

---

This handbook has been personalized for NA - ohneanmeldung
If you click "Continue", you will see the individual rows being imported. After you have verified that everything is being imported by clicking on "Next" a few times, hit "Finish". The import ends when 57 appears as your value in A1. B1 with "DIV/0" ensures that the process is aborted.

Take a look at the overview that you created in a blank worksheet earlier. You can see that all 56 data records have been imported into Palo.

### 7.3 Additional Import Options

Up until now, you have learned how to use the Import options provided by the "Data Import Wizard". However, in certain cases this solution will not be sufficient. Let’s take a look at alternatives.

#### 7.3.1 Importing from Excel-worksheets

In many cases, only few data or few Excel-worksheets are to be imported. In those cases, it does not make sense to first export these to a text file to be imported into Palo afterwards.

We can also transfer the data from an Excel source into a cube. The Excel tables just need to be adapted accordingly.
Let's assume that you want to import data, which are in the "metadata.txt" file in the folder C:\Documents and Settings\ All Users\(Shared )Documents\Jedox\Palo. Data is to be imported into the "Sales" cube of our demo database. Open the "excelimport.xls" file which is to be modified to enable import into Palo.

You will remember that each cube has a multi-dimensional, hierarchical structure. Therefore, you have to decide on a cube itself and where in the cube the data are to be inserted.

Proceed as follows:
1. Insert six blank rows in an Excel-worksheet.
2. Enter localhost/Demo in cell B1.
4. Enter the following information into the view via the "Paste Elements" function: Actual, Czech Republic, 2002 and Units. The procedure is illustrated using the example of "Actual".

A: Insertion of "Actual"
B: Result:

![Excel spreadsheet screenshot]

C: After insertion of Actual, Czech Republic, 2002 and Units

![Excel spreadsheet screenshot]

The cells just inserted are necessary to define the target of the following import.
Now we need a routine that transfers these data from C9:N13. Consequently, we copy the original data to row 17. Row 15 only serves as a visual separator.

You will remember that we used menu item "Paste Data Function" to import text files. It enabled us to easily determine the correct coordinates of PALO.SETDATA().
Select cell C18 now, and use "Paste Data Function" in the Palo menu.
After clicking "Paste" under "Paste Data Function", you see two things:

1. A value already exists in cell C18. Obviously this is because you are currently working in the demo database and values do already exist.
2. When you click the Formula Bar of Excel, you can see the appropriate PALO.DATAC function.

If you click behind the parentheses in the Formula Bar, you will see the references that the Palo formula has made.
Once you are convinced that the **PALO.DATAC** function addresses the correct cells, you can copy the formula into the other column cells. The $ character ensures that the formula is copied correctly.

Copy the contents from C18:C22 into the columns D to N accordingly. You will receive an error message. After copying, some cells contain "#NAME?".

Overwrite the months "Mrz", "Mai", "Okt" and "Dez" with "Mar", "May", "Oct" and "Dec". Please refer to row 17 in below screenshot.

The result appears as follows:

```
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost</td>
<td>Demo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sales</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Actual</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Czech Republic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Units</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Desktop L</td>
<td>527</td>
<td>473</td>
<td>732</td>
<td>427</td>
<td>760</td>
<td>400</td>
<td>259</td>
<td>646</td>
<td>717</td>
<td>668</td>
<td>577</td>
<td>571</td>
</tr>
<tr>
<td>9</td>
<td>Desktop Pro</td>
<td>778</td>
<td>388</td>
<td>543</td>
<td>211</td>
<td>73</td>
<td>573</td>
<td>638</td>
<td>357</td>
<td>723</td>
<td>559</td>
<td>798</td>
<td>144</td>
</tr>
<tr>
<td>10</td>
<td>Desktop Pro XL</td>
<td>689</td>
<td>542</td>
<td>255</td>
<td>605</td>
<td>599</td>
<td>535</td>
<td>145</td>
<td>752</td>
<td>61</td>
<td>263</td>
<td>404</td>
<td>265</td>
</tr>
<tr>
<td>11</td>
<td>Desktop High XQ</td>
<td>756</td>
<td>57</td>
<td>646</td>
<td>712</td>
<td>14</td>
<td>130</td>
<td>270</td>
<td>249</td>
<td>264</td>
<td>475</td>
<td>737</td>
<td>493</td>
</tr>
<tr>
<td>12</td>
<td>Desktop High XQ</td>
<td>720</td>
<td>733</td>
<td>399</td>
<td>423</td>
<td>195</td>
<td>170</td>
<td>208</td>
<td>763</td>
<td>659</td>
<td>208</td>
<td>465</td>
<td>35</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Desktop L</td>
<td>251</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Desktop Pro</td>
<td>562</td>
<td>339</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Desktop Pro XL</td>
<td>450</td>
<td>106</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Desktop High XQ</td>
<td>72</td>
<td>79</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Desktop High XQ</td>
<td>155</td>
<td>66</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

You will remember that we used different spellings of the months when importing the text file. In this table, the months are in German. But Palo uses the English spelling in the dimension "Months". Overwrite the months "Mrz", "Mai", "Okt" and "Dez" with "Mar", "May", "Oct" and "Dec". Please refer to row 17 in below screenshot.

The result appears as follows:

```
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>Desktop L</td>
<td>251</td>
<td>0</td>
<td>#NAME?</td>
<td>125</td>
<td>#NAME?</td>
<td>150</td>
<td>196</td>
<td>322</td>
<td>605</td>
<td>#NAME?</td>
<td>231</td>
<td>#NAME?</td>
</tr>
<tr>
<td>19</td>
<td>Desktop Pro</td>
<td>562</td>
<td>339</td>
<td>#NAME?</td>
<td>677</td>
<td>#NAME?</td>
<td>333</td>
<td>663</td>
<td>204</td>
<td>143</td>
<td>#NAME?</td>
<td>231</td>
<td>#NAME?</td>
</tr>
<tr>
<td>20</td>
<td>Desktop Pro XL</td>
<td>450</td>
<td>106</td>
<td>#NAME?</td>
<td>0</td>
<td>#NAME?</td>
<td>471</td>
<td>310</td>
<td>121</td>
<td>156</td>
<td>#NAME?</td>
<td>231</td>
<td>#NAME?</td>
</tr>
<tr>
<td>21</td>
<td>Desktop High XQ</td>
<td>72</td>
<td>79</td>
<td>#NAME?</td>
<td>335</td>
<td>#NAME?</td>
<td>180</td>
<td>356</td>
<td>183</td>
<td>236</td>
<td>#NAME?</td>
<td>231</td>
<td>#NAME?</td>
</tr>
<tr>
<td>22</td>
<td>Desktop High XQ</td>
<td>155</td>
<td>66</td>
<td>#NAME?</td>
<td>0</td>
<td>#NAME?</td>
<td>99</td>
<td>291</td>
<td>63</td>
<td>113</td>
<td>#NAME?</td>
<td>231</td>
<td>#NAME?</td>
</tr>
</tbody>
</table>
```

We already saw at the beginning of the copying procedure that the tables have differing values. The table above contains the values to be imported. The table below displays the values that currently reside in the cube. Since we haven't encountered any other issues, you can now convert the **PALO.DATAC**
function into a **PALO.SETDATA** formula. With that, you write the values from the table area C9:N13 into the area C18:N22 and consequently into the Palo cube. This procedure was already described in the "Calculated Elements (Push Rules)" section of the "Working with the Modeller" chapter.

The two Palo formulas are largely identical. The **PALO.SETDATA** formula only needs two additional parameters to describe what should be written in the cube:

First, there is the value from the import area, for example C9.
Second, the instruction to splash, i.e., "False" in this case, is required.

Select cell C18, and change the formula as follows:

\[
=\text{PALO.DATAC}($B$1,$B$2,$B18,$B$4,C$17,$B$5,$B$3,$B$6)
\]

becomes:

\[
=\text{PALO.SETDATA(C9,FALSE,$B$1,$B$2,$B18,$B$4,$B$5,$B$3,$B$6)}
\]

The data in the cube are changed immediately, and **SETDATA** returns the new value as the result. Copy the **SETDATA** function into the other cells of the table. Now both tables display the same values, and data from the original Excel table are now in the Palo cube.

You can check this, by changing the function **PALO.SETDATA** back to the function **PALO.DATAC** in the region C18:N22.
7.3.2 Importing via External ETL Tools

In addition to the import options described, you can also use external ETL tools. Here are three of the most commonly used:

7.3.2.1 Palo ETL Server

The Palo ETL Server allows the extraction of data from heterogeneous sources, as well as the transformation and the loading of inventory data and motion data into Palo-models. This makes it easy to implement a flexible data-import, which can be executed automatically.

All common relational database systems can be used as data source through a standardized interface. Even complex transformations and aggregations can be modeled with the Palo ETL Server.

The Palo ETL Server is, unlike other ETL-tools, specifically tailored to the requirements of Palo. Palo OLAP Server is supported and highly performant as both source and target system. The integration is not limited to the pure data-import from the ETL-process: the reverse direction, i.e. the "Drill-Through" of Palo into the source systems, is supported as well. It allows for the source data to be displayed at a high detail level and with that, provides an answer to the frequently asked question about the data's origin.

The Palo ETL Server is Open-Source software. Consequently, you can download and use the basic version of Palo ETL Server from our website, at no cost. More information can be found here: http://www.jedox.com/en/products/palo-for-data-integration/Introduction.html.

7.3.2.2 Cubeware

Palo is entirely supported by the Cubeware Importer. It enables fast and simple cube setup as well as the filling of cubes from several data sources.

If you are interested in the Cubeware Importer, you can find additional information at: http://en.cubeware.de/products/cubeware-product-portfolio/cubeware-importer.html.

Note: it is subject to charges.

If you do not need the complete range of functions of the Importer, you can first obtain a free tool with the name "IMP Palo" from Cubeware. You can find more information at: http://www.imppalo.com/.

7.3.2.3 Kettle

Kettle is an abbreviation for "Kettle E.T.T.L. Environment". It is considered a data integration tool and was developed to simplify the reading, transforming, transferring and loading of data from various sources. It uses a graphic interface.

If you are interested in this tool, you can find additional information at: http://kettle.pentaho.org/.
7.4 Errors during Import

You will remember that errors were contained in “Testimport.txt” file in the “Importing Data Cells” section. Besides differing months, it also contained a description “Planned” contrary to “Budget” used in our Palo model. This error was eliminated during the setup of the import.

In addition to such clearly recognizable deviations, there are also some that are not so obvious. For example, a data record would not be imported if the name of an element had additional blanks at the end. You can log such non-conforming data records using the PALO.ERROR_LOG-function. This especially makes sense for an extensive data import.

To use this function, use the file “Testimport.txt” again. You can find it in the directory: C:\Documents and Settings\All Users\Shared Documents\Jedox\Palo. An excerpt of this file will suffice for this particular example. Open the file using Excel. Delete all data records after “Austria”. Change the name “Planned” to “Budget”, and save this file under a new name in the same directory. Name this second cube “Testimport-short.txt” for example.

Because the cube "Sales" has already been filled, create a new cube using the "Cube Wizard" of the Modeller. Name it "Trial", for example, and assign the same dimensions as for the "Sales" cube.

Restart the import via the "Data Import Wizard". Select "Testimport-short.txt" as your file, and select "tabs" and "no headers". Double-click "Next", and you will get the first data record. Abort the import, and call the associated "Palo.Datac" function via the "Paste Data Function" dialog from Palo. Change the part "Datac" into "Setdata", and provide the following parameters (F1,FALSE). Your worksheet should then look as follows:

Function in F3:

=PALO.SETDATA(F1,FALSE,"localhost/Demo","Market",D$1,$C$1,$A$1,$E$2,"Units",$B$1)

Activate cell F5 and click "Insert/Function", then select category "Palo" and afterwards the desired function PALO.ERROR_LOG(). The following screen appears:
When you click "OK", you access the entry fields for this function. The parameters are:

- **Error**: the cell, which contains the error, i.e. the cell with the "Palo.Setdata" formula
- **Path**: the path to the log file and the file name (Without detailed path the error file will be stored in the folder of the import file).
- **Value**: the cell, which contains the error value, e.g. F1.
- **Cube**: the actual cube name, e.g. Market.

**Tip**: Since the error is written by the first argument "error" into the error log file and "value" and "cube" have the same behavior as the arguments of the coordinates, we can start with the first coordinate in "value". The advantage of this procedure is that we have records which are identical with the original file "Testimport-short.txt". Then it is possible to correct the errors in the error file and we can import the corrected error file with the same import procedure as before. In this post-reading we can focus our view on the records, which have produced errors in the first import sequence. Records without errors were written successfully into the cube in their correct cells by the first import sequence.

If we enter "value" and "cube" correctly as arguments, then we have to change the columns in the error file to get the same structure as in the original import file.
In order to get an optimal error message file for the remaining import, enter A1 for "value" and enter the further arguments as shown in the following screenshot:

If we want to use the error message file for the remaining import, then we need the value of F1 in that file.

If all entries have been properly made, then the cell F5 contains the following function:

\[-\text{PALO.ERROR\_LOG}(F3,"\text{error.txt"},A1,B1,C1,D1,E1,F1)\]

**Important note:** Setup of PALO.ERROR\_LOG() is actually finished at this point. However, there is one more problem and it originates in Excel. In some Excel versions, the data record following the one containing the error will be logged. Consequently, the data record from April would be displayed in the "Error.txt" file. To prevent this, click the "Palo.Setdata" again after creation of the PALO.ERROR\_LOG-Function und close the retrieved cell (in this case F3) by pressing "ENTER". This time the correct error data record will be logged. Now restart Data Import Wizard, and click "Next" a few times.
You can see in the example below that a non-confirming data record is displayed in March (Mrz). This data record was not processed, but instead rerouted to the "Error.txt" file assigned by you. Verify by opening this file.

Now you can:

- Correct the non-confirming data records as described in the "Importing Data Cells" section with respect to months.
- Open the "Error.txt" file in Excel, and correct the spelling of the months. After you store the file again, you can carry out the import for file "Error.txt", previously carried out for "Testimport-short.txt". Because this file only contains the data records previously rejected, the end result must now correspond with the result expected in the first place.
7.5 Exporting Data

Palo can export data from a cube into a CSV file. Select the Modeller, and click "Export data" with a right-click.

Then select the elements or dimensions that you want to export.
Use the query and the selection fields to restrict the export further. Blank cells can be excluded from the export. It is also possible to export base elements only. This can make sense if you want to read the data structures into a cube again. Then the base elements would be consolidated once again. Therefore, these additional consolidations are not absolutely required for the export.

If you also want to export rule based cell values, set the checkmark. If this rule values should be ignored if they have the result 0, do not use the option "Ignore empty cells" (which checks against the actual, non-rule-cell value), but set at "Query" the condition ">= 0". Note that this condition then is applied for all cells, including cells without rules.

In the next step, determine the location and name of the file to be saved. In this example, the file name "Testexport.csv" in the folder C:\Documents and Settings\ All Users\(Shared )Documents\Jedox\Palo was selected.
The CSV file (CSV = character separated values) is then created accordingly. Instead of the file extension ".csv", you can also use ".txt" and you can set a few other parameters. The following characters can be used as separators:

1. Tabulator
2. Comma
3. Semicolon
4. Blank
5. User-defined (e.g., |, # or others)

**Note:** The export can also be attached to an existing file. In this case, you must select the corresponding field ("Append at the end of file").

Then execute the export command by clicking "Finish".

When you then open the created file with Excel, the result appears as follows:

In this example, the export was not created for a CSV file, but for a TXT file instead.
With txt files, the open procedure takes several steps, and you have better control over the individual columns. In particular, you can switch the display of the decimal separator from "." to "," by clicking "Advanced" (step 3 of the open dialog).

Values with a decimal point in the text file are interpreted as text and consequently are aligned left-justified (cf. column G)
8 Managing Databases

8.1 Introduction

You have worked with the demo database up until now. This is an existing model. Of course, you can also create, manage and delete your own databases using Palo.

In order to create a new database, click on "Modeller..." in the Palo-menu. Then you click on "Start Palo Wizard" (red arrow), in the following dialog-window:

![Palo Wizard dialog](image)

You will get the "Palo Wizard"-dialog:
Using the Palo Wizard, you can:

- Create new databases
- Create databases from templates
  (This item is planned for a later Palo version and consequently cannot be activated now)
- Delete existing databases
- Register a new server
- Unregister a server

If the "Palo Wizard" has not yet established a connection to the server, the connection button displays "Connect". Otherwise it displays "Disconnect".

The "Select server" menu item is important when working in networks and with user rights.

You can find explanations in the chapters:
- Installing and Deinstalling Palo in the "Working in Networks" section
- Working with User Accounts in the "Applying User Rights" section

### 8.2 Managing Databases

In order to activate the database administration, you must have a connection to the server. If it is not yet established, click on "Connect" (red arrow):
8.2.1 Creating New Databases

Once a connection has been established, you can create a new database. Select the corresponding item, and click “Next”.

In the next step you assign a name to the new database. Let’s assume that it should be called “Demo”. The name was not chosen arbitrarily, we would like to demonstrate something.

If you intend to use a name that already exists, you will receive an error message. Palo reminds you that “Demo” already exists.
Click “OK” to return to the Palo Wizard to assign another name, e.g., "New test", again you receive an error message, this time due to invalid characters.

The character not allowed here is the blank between New and test. Click "OK" to assign another name again. If you for example use “Test”, “Test1” or Test30.05.2009”, the database will be created immediately.

Clicking “Next” results in a message that the database has been created successfully.

The Palo Wizard closes when you click “OK”.

The database is now ready for use. You can open the "Modeller", in order to create dimensions and elements, as well as to combine dimensions in cubes.

**Tip:** You know that cubes, dimensions and elements define data structures in Palo (The Modeller chapter). You also know that the respective structures only apply to defined cubes. As a result, subsequently created dimensions are not contained in an existing cube. To include these, you would have to delete a cube and set it up again, however, the data are lost. We therefore recommend that you create all required structures **before** saving data in a Palo database.
8.2.2 Deleting Databases

Of course, you can also delete existing databases. The first steps are the same as to create a database. When you see the following screen, select "Delete an existing database"

If you click "Next", you will get a selection of databases. Select your new database "Trial".

Make sure that you select the correct database. Then click "Finish".

This handbook has been personalized for NA - ohneanmeldung
You must confirm whether you want to delete this database.

![Delete Database dialog box]

You will then get a message about the successful deletion of this database. You must confirm this note by clicking "OK".

![Database successfully deleted dialog box]

Thereafter, the "Palo Wizard" will be closed.

Alternatively, you can stop the PaloServerService and then delete the desired database folder in the \...Palo\data folder (see also chapter Backup of Databases)
9 Application Example

9.1 Part 1: Developing of a Palo Cube

This example is divided into three parts. The Palo cube is created in the first part, but since we plan to develop a new application, the first step is to create a database. Please also refer to the preceding chapter for instructions.

9.1.1 Creating Databases

Our example is called "Sample". Open a blank Excel-workbook. Click Palo, and select the "Palo Modeller". Then click on "Start Palo Wizard..." to open the "Palo Wizard"-dialog:

Select "Create new database". Leave the server settings at "localhost", and click "Next".

Enter the name "Sample" next to database in the following dialog window, close by clicking "Finish". You will then get a message that the database has been created successfully. Confirm with "OK". The Palo Wizard will then close. The database that you just created can now be used.

In the next step, you can now define the necessary parameters in the Modeller.
9.1.2 Selecting Databases

You know that the Modeller serves as a tool to create and manage data structures with the help of cubes, dimensions and elements. Open the Modeller, and select the database “Sample”.

You will get a blank view, and you can enter the new dimensions.
9.1.3 Creating Dimensions

We need the dimensions Projects, Resources, Months, Days and Measures. Click on the text ("Hit RETURN to add Dimension…") and press "ENTER".

1. Creating a new dimension

The first new dimension is created and marked to be renamed (picture 1). Name this dimension "Projects". Then double-click "ENTER". After the first click, the dimension name is created (picture 2).

Another "New dimension" is created with the second click. Name it "Resources". Create all further dimensions accordingly. Finally, you can see the just created dimensions in the Modeller (to the right).
9.1.4 Creating Elements
Once you have dimensions, you can create the respective elements. If necessary, they should be consolidated and then displayed hierarchically.

9.1.4.1 Projects
For this dimension you need the structure displayed below:

All Projects

- Internal
  - 001 Website
  - 002 Flyer

- External
  - 101 Smith & Co.
  - 102 Mayer Ltd.
  - 103 South Corp.

First create all elements displayed above.

Click the dimension “Projects” in the Modeller:

Hit “Return”, and you get a new element. Assign the name “All Projects”.

Hit “ENTER” twice. First hit will change the name, second hit will create a new, additional element.

Assign the names you see on the right:
Now the elements must be consolidated. Select "Internal". Then click "Consolidate elements" at the bottom of the window.

Select the two elements 001 Website and 002 Flyer and add them to the "consolidated elements" window using the blue arrow.

You will see the following window in the next step:

Close the procedure by clicking the green checkmark.
Consolidate the elements "101 Smith & Co.", "102 Mayer Ltd." and "103 South Corp." to "External" in the same way. Then consolidate "Internal" and "External" to "All projects". In conclusion, display the elements hierarchically:

Compare it with the structure described in the beginning:

All Projects

• Internal
  • 001 Website
  • 002 Flyer

• External
  • 101 Smith & Co.
  • 102 Mayer Ltd.
  • 103 South Corp.

9.1.4.2 Resources

Now create the elements of the dimension "Resources" in the same way.

Create the base elements Frank, Bill, Mike, Tom and Bob and consolidate them to "All Resources".

Click "Resources" to activate the dimension. Create a new element by hitting "ENTER". Assign the name "Frank". Press "ENTER" twice. Hit "ENTER" twice. First hit will change the name, second hit will create a new, additional element.

The following screen is displayed at the end:

We created "All resources" as the last one, move it using the arrow buttons to the first place. You see the following screen:

The employees should now be consolidated to "All resources". This time do not consolidate via the function at the bottom of the window, instead use the context menu (right-click on "All resources").
Next select all employees and move them to the right using the blue arrow.

You will see the consolidated elements in the right window. Close the consolidation by clicking the green checkmark.

Display the dimension hierarchically. You will see the structure described in the beginning:

9.1.4.3 Months

Create the months via the "fill" function in Excel and copy them to Palo, i.e. open a blank Excel-worksheet, and enter the months "Jan", "Feb" and "Mar" in cell A1:A3. Mark these cells, and drag the marking down to cell A12.

Now highlight these months, and copy the marked section using the context menu.
Open the Modeller, and click “Months”.

Hit Return to activate the area "Elements". Palo allows you to paste elements via the context menu.

"Quarters" and "Year" are still missing. Create them in the now familiar way, i.e. click the last element and press ENTER. A new element is created. Name it: Qtr. 1, Qtr. 2, Qtr. 3, Qtr 4 and Year.

You can see the result below:

Now perform the corresponding consolidations. Check the result using the hierarchical display:
9.1.4.4 Days

The days are best created via the "Fill" function from Excel. Prepare by providing a total of 31 days.

Create days

The days are copied into the dimension "Days". Use the copy command of the context menu in Excel.

Then open the Modeller. Click the dimension "Days" and paste the elements via the context menu.

You obtain the following result:
Add an element "All days", i.e. click element "31", and press Return. Name the new element "All days". Then consolidate all days in this element.

You can see the hierarchical display on the right:
9.1.4.5 Measures

This dimension is composed of the two elements "Actual hours" and "Budget hours". Create these elements:

- Actual hours
- Budget hours

9.1.5 Creating a Cube

Click the "Database" button after you have filled the last dimension with elements. Then you return to the screen with the existing dimensions. Alternatively, you can click "Database card".

The next window displays the dimensions that you created, i.e. Projects, Resources, Months, Days and Measures.

A click on "New cube" will open another window.
Enter "Timesheet" next to name. Assign the listed dimensions to the cube by selecting them. Click "Projects" and keep the shift key pressed down, then click "Measures". You will see the selected dimensions as shown below:

![Cube Creation Wizard](image)

Now click the blue arrow pointing to the right. This is how you assign dimensions to a cube. Click "Finish" to conclude the procedure.

The cube is created. You can see the result in the screen below:
If you, after the completion of the cube, want to change the name of the dimension, then open the context menu by a click on the right button of the mouse:

```
Dimensions:  Cubes:
Projects       Timesheet
Resources
Month:
Days:          New dimension
Measure:       Delete dimension...
               Rename dimension
               Edit dimension
               Export dimension...
               Dimension Information ...
```

It sometimes happens that companies change their department or area names. In such cases, it makes sense to be able to change a name without having to reinstall all data.

The first task is now solved. End the Modeller by clicking "Close".
9.2 Part 2: Creating Palo Worksheets

Now let's get to Part 2. We develop an application with the newly created cube.

9.2.1 Creating Project Reports

Open a blank Excel-worksheet. Retrieve the required structures from Palo using the options you know, e.g., Paste View, Paste Element and Paste Data Function.

9.2.1.1 Inserting Structures from Palo

Open the "Paste View" dialog.
Select the dimension "Projects" for Row titles and "Measures" for Column titles.

Close the dialog by clicking "Paste".
The following screen appears:

![Excel screen with sample data]

Open the consolidated element "All projects" with a double-click. Proceed in the same way for the two subsequent consolidated elements. Then the application appears as follows:

![Excel screen with project data]

Move the consolidated elements downward. Click row 8 to do that. Position the mouse pointer on the element, so that a crosshair appears. Hold the **SHIFT KEY** and the left mouse button down and at the same time drag the row down to the line between rows 15 and 16.
The elements should be arranged as follows:

1. Moving "All projects"

2. After moving "All projects"

The element "All projects" has now been moved into row 15. At the same time, the elements above 15 have been moved from rows 9 to 15 (old) to rows 8 to 14. In other words, the shift key combines two procedures. The cut and paste of the data belonging to "All projects" (row 8 to row 16) and the subsequent delete of the now blank row 8.

Proceed in the same way with the elements "Internal" and "External". Move them, so that they are inserted above "All projects". The screen shown to the right appears:
Select the area E7 to I7. Click "Paste elements" and select "Resources". Open the "Select All" dialog, and mark the 2:
Your screen will look like this:

```
      E7       
7 |------------------------------------|
<table>
<thead>
<tr>
<th>A    B   C   D        E       F    G    H    I</th>
</tr>
</thead>
<tbody>
<tr>
<td>localhost/Sample</td>
</tr>
<tr>
<td>Timesheet</td>
</tr>
<tr>
<td>All Resources</td>
</tr>
<tr>
<td>Jan</td>
</tr>
<tr>
<td>All Days</td>
</tr>
</tbody>
</table>
7 |------------------------------------|
| 8  | 001 Website   | 0.00 | 0.00 | Frank | Bill | Mike | Tom | Bob |
| 9  | 002 Flyer     | 0.00 | 0.00 |
| 10 | 101 Smith & Co.| 0.00 | 0.00 |
| 11 | 102 Mayer Ltd.| 0.00 | 0.00 |
| 12 | 103 South Corp.| 0.00 | 0.00 |
| 13 | Internal      | 0.00 | 0.00 |
| 14 | External      | 0.00 | 0.00 |
| 15 | All Projects  | 0.00 | 0.00 |
|------------------------------------|
```

Reduce the size of column D. Then click field E8 and select the "Paste Data Function" Palo dialog. Make sure that the cube "Timesheet" is selected and also "Guess Arguments". Then click "Paste".

This handbook has been personalized for NA - ohneanmeldung
Copy the formula to the range E8:I15. Your screen then looks as follows:

![Excel screenshot with formulas entered]

9.2.1.2 Inserting Charts
Select the area A7:C12. Insert a Column graph via Insert/Chart.
Adapt this chart to range A16:I30.
The following screen appears:
9.2.1.3 Entering Planned Hours

Enter the number of planned hours for the individual projects in column C ("Budget hours"). Remember that values to be splashed must begin with the special character ".#. Enter the following values: 800, 400, 600, 700 and 400. The corresponding results in the chart change immediately.

You will see the following:

Now switch the dimension "Year" to "Jan", i.e. double-click "Years", open "Qtr.1" and then select "Jan".
Please note how Palo breaks down the annual values of each project to get a monthly value. The chart immediately adapts as well.

Select the area C7:C15. Click the “Format Painter” icon from Excel (brush). Select E7:I15. The format is applied automatically. Your screen then looks as follows:
9.2.2 Creating Time Entries

Open a new Excel-workbook and create a time entry sheet for current project hours of staff members. Call the Palo "Paste View" dialog. Move the dimension "Projects" into the Column titles and move "Days" into the Row titles. After you click "Paste", you see the following screen:

```
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>localhost/Sample</td>
<td></td>
</tr>
<tr>
<td>Timesheet</td>
<td></td>
</tr>
<tr>
<td>All Resources</td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td></td>
</tr>
<tr>
<td>Actual hours</td>
<td></td>
</tr>
<tr>
<td>All Projects</td>
<td>0.00</td>
</tr>
<tr>
<td>All Days</td>
<td>0.00</td>
</tr>
</tbody>
</table>
```

Expand the display to make "Internal" projects visible. Only the consolidated element is visible for "External". Double-click the consolidated element "All days" to get the individual days. Finally, delete row 8 ("All days"). Your sheet should now look like this:

```
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>localhost/Sample</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Timesheet</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All Resources</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Year</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Actual hours</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All Projects</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All Days</td>
<td>0.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>6</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>
```
9.2.2.1 Entering Times

Set "Resources" to the first staff member.

Set the dimension "Year" to the month "Jan" again. Your sheet should now look like this:

Set the dimension "Year" to the month "Jan" again. Your sheet should now look like this:
Frank can now enter his "Actual hours". Enter Frank’s data for the first three days of the month. Enter the values 8, 4 and 2 under "001 Website" and values 1, 3 and 6 under "002 Flyer". Observe how the consolidated data change for "Internal" and "All projects".

You can see the result for Frank below.

Now it’s Bill’s turn. Double-click "Frank", and change the element to "Bill".

This handbook has been personalized for NA - ohneanmeldung
Enter the first three days for him as well: "001 Website" 4, 8, 6 and "002 Flyer" 4, 0, 2. Observe how the consolidated values change here too.

In reality, Bill would enter his data on another PC, but nevertheless, the result would be available immediately.
9.2.3 Transferring Data Entries to Reports

The entries for Frank and Bill have been completed. Now return to your report. You can see that the data have been transferred automatically. At the same time, the view has changed.

If you used Excel only, you would use the formula
\[ =\text{SUM}([\text{Book2}]\text{Sheet1!}D8:D38) \]
in row 8, for example. In that case, Sheet1 may only contain data for "Bill" (cf. the last view of Book2 and the subsequent excerpt).

The value in F8 is 18, which is the sum of the website hours from Bill.

You would need an identical table for "Frank". It does not matter in Palo whether the cell A3 displays "Frank", "Bill" or "All resources". The PALO.DATAC formula ensures that the correct values are written into the fields of Sheet1.

This is the magic of Palo …

…. and this completes the 2\textsuperscript{nd} part.

This handbook has been personalized for NA - ohneanmeldung
9.3 Part 3: Creating of a Web Application with Palo, Excel and Worksheet Server

The next part explains how you can create an Internet solution using a combination of Palo and Excel using Worksheet Server. You can convert any Excel-workbooks into server-based Web applications (Excel-to-Web) and hence enable you to display and edit Excel-worksheets in Web browsers. This way, you can use enterprise spreadsheets independent of local Excel installations in an Intranet or worldwide on the Internet.

9.3.1 Preparing of the Excel-Workbook

To simplify our work, we will copy the Excel Worksheets books created in the 2nd part above into one book. Open Book1. Save it as “Timesheet.xls”. Then switch to Book2. You can move the time entry sheet into the “Timesheet.xls” file via Edit/Move or Copy Sheet. Move this overview before Sheet2.

Make the following changes:
1. Change the name of Sheet1 to “Report”.
2. Change the name of Sheet1(2) to “Input”.
3. Delete the table sheets starting with Sheet3.
4. Set the dimension “Year” to “Jan” in “Report” if necessary.
Your screen now appears as follows:

The Worksheet-Server requires another change in the formatting. Select the area A16:I30, and link the cells in Format/Cells/Alignment and Merge cells here.
The first two rows of the report are irrelevant for the Web report. Therefore, hide them:

In the next step, prepare the area C8:C15 ("Budget hours") for new data entry. Mark the area C8:C15 and click Format/Cells and then register-card "Protection". Remove the checkmark in the box "Locked".

Do the same for range A3:A5.

Now switch to the sheet "Input". Hide first two rows, in addition, hide row 5 (showing "Actual hours").
Then unlock the ranges D8:E38 and A3:A4 (Format/Cells/Protection and the checkmark next to "Locked" is removed). The table looks as follows:

Now add a menu:
Use the still blank Sheet2 in this workbook. First move it in front of the sheet "Report". Rename Sheet2 "Start".

Let's create the actual menu. How about a header "Welcome to your Time Recording-application" and two simple links to both tables?

Enter "Welcome to your Time Recording-application" in cell B3. Click cell C5, and call Worksheet-Server/Insert Hyperlink.
Switch the field next to Internal Hyperlink to sheet to "Input". Then enter "Project Hours" next to Friendly Name.

Click cell C7, and call Worksheet-Server/Insert Hyperlink again. Switch the field next to Internal Hyperlink to sheet to "Report". Enter "Project Budgets & Analysis" for a Friendly Name.

Then the start-up sheet appears as follows:

Finally some cosmetic changes:
1. Reduce the size of columns A and B.
2. Format the greeting in Arial 14 and bold
3. Replace the font colour of the hyperlinks (blue) with black.

This handbook has been personalized for NA - ohneanmeldung
Your start-up sheet now looks as follows:

![Excel Sheet Image]

Your application is finished! You can now upload this application to the Worksheet Server.

### 9.3.2 Creating of the Worksheet Server Application

Click Worksheet-Server and then Save. In the following dialog, set the "Upload" folder as storage location. With the installation of Worksheet Server, the folder "kd305016355" should be set up in the directory C:\Programs\Worksheet-Server\WSS\root\folders. "Uploads" is a subfolder of "kd305016355".

**Note:** A different path may be defined for new installations of Worksheet Server: C:\Programs\Jedox\Worksheet-Server and, regardless of default settings, you may also have your individual path definition of course.

Leave "Timesheet.zip" as file name, and click Save. The data are transferred to Worksheet Server. A message displays the progress.

Worksheet Server confirms the successful transfer shortly afterward.

![Worksheet Server Confirmation]

The application is then available on the Web Server.

Browser address: http://localhost/folders/kd305016355/timesheet/index.html
Click the indicated path, and the Web application opens in a browser. First log on as Administrator (admin/admin). You might have to confirm the password.

![Worksheet-Server 2.0 Login](image)

**9.3.2.1 Setting up Users**
You can set up and manage users in the next step. You see the following screen after you click "LOGIN".

When you click "User and Passwords", you can add Bill as your new user. Enter "bill" as user name and password in the following list.

![User List](image)

Then click "Save".

---

This handbook has been personalized for NA - ohneanmeldung
The view is switched to "Edit User List".

<table>
<thead>
<tr>
<th>No.</th>
<th>Username</th>
<th>Initial Password</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>admin</td>
<td>admin</td>
</tr>
<tr>
<td>2</td>
<td>bill</td>
<td>bill</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

End this dialog by clicking "Logout" in the upper right.

It takes you back to the login screen of Worksheet Server. Log in as Bill. If you are logging in as "Bill" for the first time, you must confirm or change the password.

1. First login as "Bill"
2. Dialog for changing the password

9.3.2.2 Data Entries by Users
Next you see the page, which you defined as your start-up page in the Palo/Excel application.

Welcome to your Time Recording-application
- Project Hours
- Project Budgets & Analysis

This handbook has been personalized for NA - ohneanmeldung
You are now in the Web application. Click "Project Budgets & Analysis".

Note: If no values are displayed, you might have installed an older version of Worksheet Server. In this case, check the version by clicking Worksheet-Server/About in Excel. The following screen is displayed:

If not "108" at a minimum, install a newer version. Close Excel, and reopen the file "Timesheet.xls" after the installation of Worksheet Server. Save the file once again as "Timesheet.zip". When you now start Worksheet Server, you can see the required information.
9.3.3 Comparing the Worksheet Server and Palo/Excel

Compare the views.

Above: The "Report" sheet on the Worksheet Server.

Right: The same sheet in Excel as "Timesheet.xls".
The Web application is linked directly with the Palo Database. In the browser, switch the month from "Jan" to "Feb".

Hit "Back" in the upper left corner and then "Project Hours" in the start-up window. In the "Project Hours" page, select "Feb" and "Bill". Then you can enter data for Bill.

Enter “001 Website” 3, 5 and “002 Flyer” 8, 6 for the first two days. Observe how the consolidated fields change.
Click “Back” and then “Project Budget & Analysis” again. The data that you just entered for “Bill” are contained in the Web application:


We have reached the end of the 3rd part. You have now experienced how Palo and Excel can be used via the Internet thanks to Worksheet Server. Data are always consistent and up-to-date.


This handbook has been personalized for NA - ohneanmeldung
10 Administration of User Rights

User rights regulate the access to databases, cubes, dimensions and dimension elements. There are users, user groups and roles. These are given rights with reference to cell data as well as to selected system operations.

Rights can be given to the following areas:
- Users
- Passwords
- Groups
- Databases
- Cubes
- Dimensions
- Dimension elements
- Rights
- Cell data
- System operations (SysOp)

We shall call these areas "rights objects".

The following access rights will be used. The access rights are hierarchically ordered. Consequently, higher rights include the subordinate rights.

10.1 Hierarchy of Rights

The rights are hierarchically defined as follows:

- S (splash): Exists only for the rights object "cell data". It includes writing into consolidated cells and (indirectly) their children down to the lowest level.
- D (delete):
  Permits the deletion of rights objects.
- W (write):
  Permits the writing of rights objects.
- R (read):
  Permits the reading of rights objects.
- N (none):
  No type of access permitted.

Higher rights include the subordinate rights automatically.

The highest access right is expressed by "splash", but it refers only to the cell data. For all other rights objects, "delete" is the highest access right. The hierarchy is therefore "splash", "delete", "write", "read" and "none".
10.2 System Database

10.2.1 System Cubes and System Dimensions

The administration of users and rights takes place in the system database in system cubes and their dimensions for all databases, to which the PALOServerService has access, at first.

In order to see the structure of the system database, please invoke the Modeller and select an arbitrary database, for example "Demo".

Then click on the "User administration" icon:

![Modeller Interface]

The dimensions visible on the left are editable, the dimensions \_USER\_PROPERTIES\_ and \_RIGHT\_OBJECT\_ are not editable. The latter are shown only on the right side of the cube. Consequently, the system database contains the 4 cubes \_USER\_USER\_PROPERTIES, \_USER\_GROUP, \_ROLE\_RIGHT\_OBJECT and \_GROUP\_ROLE.

The dimensions of these cubes are:

\_USER\
- All elements of this dimension are predetermined or created users.
- The user "admin" is always present and can neither be renamed nor deleted.

Additionally create the users "Anna", "Bert", "Cath", "David", "Elke" and "Franz" here.

\_USER\_PROPERTIES\
- Contains the element "password".

This handbook has been personalized for NA - ohneanmeldung
#_GROUP_
- All elements of this dimension are predetermined or created user groups.
- The user group "admin" is always present and can neither be renamed nor deleted.

Additionally create the groups "CEO", "Head of Sales", "Head of Operations", "Management", "Employees A" and "Employees B" here.

#_ROLE_
- Contains the predetermined roles: admin (cannot be removed!), poweruser, editor and viewer.

#_RIGHT_OBJECT_
- Contains all rights objects. These are:
  user, password, group, database, cube, dimension, dimension element, cell data, rights, system operations, event processor, sub-set view, user info and rule.

All elements of these dimensions are text elements.

10.2.2 Cubes in the Database "System"

These cubes are now presented in the order of the allocation of rights:

10.2.2.1 Cube #_ROLE_RIGHT_OBJECT

We insert the following view:
At a new installation, the following rights will be assigned to the roles "admin", "poweruser", "editor" and "viewer" by default:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/System</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>#_ROLE_RIGHT_OBJECT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>ROLE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>admin</td>
<td>poweruser</td>
<td>editor</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>D</td>
<td>R</td>
<td>N</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>D</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>D</td>
<td>R</td>
<td>N</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>D</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>D</td>
<td>D</td>
<td>R</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>D</td>
<td>D</td>
<td>R</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>D</td>
<td>D</td>
<td>R</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>S</td>
<td>S</td>
<td>W</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>D</td>
<td>R</td>
<td>N</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>D</td>
<td>W</td>
<td>N</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>D</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>D</td>
<td>D</td>
<td>W</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>D</td>
<td>D</td>
<td>W</td>
</tr>
<tr>
<td>18</td>
<td></td>
<td>D</td>
<td>D</td>
<td>W</td>
</tr>
</tbody>
</table>

Where the letters have the following meaning:

- **S (splash):** Exists only for the rights object "cell data". It includes writing into consolidated cells and (indirectly) their children down to the lowest level.
- **D (delete):** Permits the deletion of rights objects.
- **W (write):** Permits the writing of rights objects.
- **R (read):** Permits the reading of rights objects.
- **N (none):** No type of access permitted.

It is also possible not to make any entry in the cell of a system cube. The meaning of an empty cell depends on the respective system cube.

In the cube `#_ROLE_RIGHT_OBJECT`, "empty" stands for the topmost right, that is, depending on the rights object, "D" or, with cell data, "S".

This handbook has been personalized for NA - ohneanmeldung
With all rights objects, the role of "admin" has the highest access rights at her disposal: "delete" or "splash" respectively. The user "admin" is firmly anchored in the system and cannot be altered. That is, the user "admin" cannot be deleted and his rights are firmly predetermined.

The roles "poweruser", "editor" and "viewer" have standard predetermined rights at first. They can be adapted to your needs. In addition, further roles can be created.

The "poweruser" has far-reaching standard editing rights. Besides writing and reading rights, these include also the right to delete cubes, dimensions and elements.

The "editor" is allowed to enter data and in addition to view data. Also he is allowed to create Rules and Subsets and to see the objects with "R" entry.

The "viewer" is allowed to read data and to see the objects with "R" entry.

10.2.2.2 Cube #_GROUP_ROLE
Build the following view of the cube #_GROUP_ROLE:

Here it is recorded which role a group gets. Adopt the entries from the image.

The user is allocated to a group by putting "1" into the cell.
The user "admin" belongs to the group "admin". This cannot be changed.
Only one role can be allocated to a group.
10.2.2.3 Cube #_USER_GROUP

Build the following view of the cube #_USER_GROUP:

Here it is recorded to which group a user belongs. Adopt the entries from the image.

A user can be a member of several groups. Then he is equipped with the highest rights resultant from it.

10.2.2.4 Cube #_USER_USER_PROPERTIES

Build the view, as seen on the right, of the cube #_USER_USER_PROPERTIES:

This cube contains the user passwords. Adopt the entries from the image.

This cube only contains passwords and has neither a direct nor an indirect effect on rights.
Users, groups and roles are administered system wide. This information has validity for all databases. These rights, granted by the system database, can be restricted further (but not extended!) for each single database.

Consequently, the system-cubes can be divided into two categories:
The system-cubes in the system-database grant the rights to all databases administered by the PALOServerService (general rights = role area).
The system-cubes in the respective database can restrict the access rights of this database further (individual database rights = data area).
These cubes are the #_GROUP_CUBE_DATA -cube and the #_GROUP_DIMENSION_DATA_<name of the dimension> - cubes.

10.2.3 Restriction of Rights in single Databases

Important: These restrictions refer to the "cell data"-rights object only – rights granted, for example, to the cube itself, are still obtained from the system-database!
In the database "Demo", additionally create the cube "Operations" with the same dimensions as in the cube "Sales". Then we have the cubes "Sales", "Market" and "Operations" in the demo-database. So far, user David has been given the rights to create and delete cubes (David is a member of the management group and "Management" has the role of "poweruser").
10.2.3.1 Cube #_GROUP_CUBE_DATA

In the "Paste View", choose the database "Demo" and the "User management cubes":

By the entry R (as seen on the right), the group "Management" has only read-access to the cube "Operations". This refers to the cube cells only. As a "poweruser", David is still allowed to delete or empty the "Operations"-cube on the Modeller-level!

The previous writing-rights for the group "Employees A" for the cubes "Sales" and "Operations" are withdrawn.

The previous reading-rights for the group "Employees B" for the cube "Sales" are withdrawn.

---

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>localhost/Demo</td>
<td>CUBE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#_GROUP_CUBE_DATA</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GROUP</td>
<td>Sales</td>
<td>Market</td>
<td>Operations</td>
</tr>
<tr>
<td>admin</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>poweruser</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>editor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>viewer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CEO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Head of Sales</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Head of Operations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management</td>
<td></td>
<td></td>
<td>R</td>
</tr>
<tr>
<td>Employees A</td>
<td>N</td>
<td>R</td>
<td></td>
</tr>
<tr>
<td>Employees B</td>
<td>N</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
10.2.3.2 Cube #_GROUP_DIMENSION_DATA_<Name of dimension>

This system cube contains the link of the system dimensions #_GROUP_ and a dimension within a database. This cube can restrict the right given by the role-area, but it cannot expand the given right. The cell can contain the values D, W, R, N and blank.

The following access rules apply:
The children of a consolidated element have the same access rights as their parent unless another right has been assigned to them explicitly. If an element has two parents, the higher right rules.

Example A – further restrictions:
The group "Employees B" is not allowed to read the cell data of Qtr. 1:

The entry N is inherited from Qrt.1 to Jan, Feb and Mar.

Example B – "Non-extension" of rights:

By the entry D in the screen on the left, members of the group "Employees B" can neither delete nor write, just read, because this entry cannot extend the rights given to the "viewer"-role.
10.2.3.3 Cube #_CONFIGURATION

In this system cube, you can specify whether someone who has not the right to see certain cell entries has the permission to see the corresponding element name or not.

Example:

The group "Employees B" has not the right to see the cell entries of Qtr. 1, Jan, Feb and Mar.

Without a change in the cube #_CONFIGURATION a member of the group Employees B can see the element names QTR. 1, Jan, Feb and Mar when creating a view and can use them for the view. In the view itself will then only the corresponding cell entries are not shown.

In order for hiding an element name if it has the entry N, we go on the button "User management cubes" and select the cube #_CONFIGURATION.

Then we enter "Y" under HideElements:

Thereafter, elements without cell-read permissions are hidden during view creation:
10.2.4 Additional Rules

Effects of Rights on Calculations
Rights of any kind never affect computations or consolidations. For example, a "none" right in the cube #_GROUP_DIMENSION_DATA_<Name of dimension> does not influence the calculation of dependent contents in consolidated cells.

Splashing
You can only splash if the role is set to "S" for cell data and there are writing rights (W) for #_GROUP_CUBE_DATA and #_GROUP_DIMENSION_DATA_<Name of dimension>. The splash right (S) is only assigned in the #_ROLE_RIGHT_OBJECT cube and not in #_GROUP_CUBE_DATA or #_GROUP_DIMENSION_DATA_<Name of dimension>.

Consolidated Elements
Super ordinate elements pass on rights to subordinate ones as long as the latter do not have any rights defined of their own. Consequently, "splash" rights make it possible to change cell data of consolidated cells directly. At splashing (i.e., writing in consolidated elements), "splash" rights are dominant in the sense that the rights of children are being ignored. If an element below a splashed element has "read" rights, this is ignored during the splash procedure. But splash rights do not dominate for pure reading rights. Any possible "none" rights below "splash" rights will be respected. In other words, as long as a user has writing rights for the current consolidation level and otherwise splash rights, he may splash. If he only has reading rights for the current consolidation level, then he may not. The right "N" below that, however, would be factored in. Consequently, the combination of splash and the rights of the consolidated element in question is the deciding factor.
Example
User "David" is a member of the group "Management" and thus a poweruser. Therefore he has the right to splash all cell contents. Within the group "Management", we have the dimension element "Year", which is consolidated from "Jan" to "Dec". In this case, writing-rights for the group "Management" for the element "Year" have the effect, that the splash command can be executed, irrespective of which rights are granted for the elements "Jan" to "Dec". The data-cells addressed by these elements will therefore be modified by the splash command.

Rule: W+Splash breaks R:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>#_GROUP_DIMENSION_DATA_Months</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Management</td>
<td>Employees A</td>
<td>Employees B</td>
</tr>
<tr>
<td>4</td>
<td>Year</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Qtr. 1</td>
<td>W</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Jan</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Feb</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Mar</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Qtr. 2</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Qtr. 3</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Qtr. 4</td>
<td>R</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Members of the group "Management" no longer have the right to splash for the consolidated elements "Qtr 1" up to "Qtr 4", because they only have reading-rights for them.

Rule: R+Splash = no right to splash.

Rights for Attributes (Attribute-Dimension, Elements, Data)
Independent of what has been said so far, only the following rule applies to attribute-dimensions and -cubes:
Only those who have the right to modify dimension elements also have the right to add or delete attribute-fields and to write attribute-cube data. In particular, rights-cubes shall not be created for the attribute-dimensions nor shall the attribute-cubes be listed in the system-dimension #_CUBE_.

Users without a group role: A user cannot log in, if he is not a member of a group with a role.

Example A:
The assignment of roles of "Management" and "Employees A" are temporarily removed. Then, David, who is a member of these groups, cannot log in any more.

Example B:
For the group "Management", the role "poweruser" is assigned again. David belongs to the groups "Management" and "Employees B". Then, David can log in again, because he belongs to a group with a role.
Modifying rights during a session
The server responds dynamically, if allocations, which influence the rights of a user, change during a session.

Example A:
David belongs to the groups "Management" (poweruser) and "Employees B" (viewer). At the time David logs in, the groups have the roles given in braces. During David’s session, the role-assignments of the two groups are cancelled, so that both groups don’t have any role any more. With the next inquiry by David, PALO will deny the access.
(Please undo all changes of role-assignments, which have possibly been performed in this example!)

Example B:
David, who is a member of the group "Management" as well as of the group "Employees B", has logged in as a user. Thereby he has the highest rights from both groups which are the rights as a "poweruser". Therefore he has the right to splash all cell contents. Now he is removed from the group "Management" by the "administrator". He only remains a member of the group "Employees B". Access as "poweruser" will immediately be deprived of David. He can only continue his work as a "Viewer".
(Please undo all changes of role-assignments, which have possibly been performed in this example!)

10.2.5 Applying User Rights
The setup of users and managing their rights is completed with that. Now you have to apply them. In other words, you must ensure that a user has to log on with his username and password. Afterward, he should only be able to work in line with his rights assigned.

10.2.5.1 Creating Server Entries for New Users
Check the user rights function for "Elke". Click the "Palo Wizard" in Palo. Select "Register new server", and confirm by clicking "Next".

This handbook has been personalized for NA - ohneanmeldung
Enter the following information in the next dialog:
"Test_Elke", localhost, 7777, "Elke" is the username, "elke1" is the password.

Click "Test connection". A message appears stating that the test was successful.

If you received an error message, you might not have spelled the user names and/or password in the same form as stored in Palo. Pay special attention to upper and lower case letters.

Confirm the message with "OK". Then click "Finish". You receive confirmation that the server has been registered.

The Palo Wizard closes when you click "OK".
When you now return to the Modeller, you see that "Test_Elke" was created. Click the icon for "Disconnect". The display changes to "Connect".

![Modeller interface](image)

Then you can click "Connect".

An entry menu appears.

![Entry menu](image)

Enter "Elke" as user name and "elke1" as password.
Open the "Paste View" dialog again.

Please note: Elke can view the structures of all cubes. However, because of the rights restricted through "#_GROUP_CUBE_DATA", she is only allowed to write to the base-cell data of the cube "Market". She is allowed to view, but not to modify, the values in the cube "Operations" and she is not allowed to view the values in the cube "Sales".

Compare the two displays for the cubes "Sales" and "Market".

Note: A special feature of writing rights must be noted in this context. Writing rights for cell data permit writing of any number in principle. But there is one exception, and that is the number "0". Entry of zero is not interpreted as a number in this case, but instead as "delete". But editors do not have "delete" rights. This results in an error message.
Example:
Elke tries to replace her entry "100" with "0".

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Test_Elke/Demo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Market</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Jan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2002</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Germany</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Actual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Units</td>
<td></td>
<td>Gross Profit</td>
</tr>
<tr>
<td>9</td>
<td>All Products</td>
<td>100,00</td>
<td>-1,608,66</td>
</tr>
<tr>
<td>10</td>
<td>Stationary PC's</td>
<td>100,00</td>
<td>-1,608,66</td>
</tr>
<tr>
<td>11</td>
<td>Desktop L</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Desktop Pro</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>13</td>
<td>Desktop Pro XL</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>14</td>
<td>Desktop High XL</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>15</td>
<td>Desktop High XQ</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>16</td>
<td>Server Power XC</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>17</td>
<td>Server Power TT</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>18</td>
<td>Server Dual C</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>19</td>
<td>Server Dual XC</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>20</td>
<td>Server Lion RX</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>21</td>
<td>Portable PC's</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>22</td>
<td>Monitors</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>23</td>
<td>Peripherals</td>
<td>0,00</td>
<td>0,00</td>
</tr>
</tbody>
</table>

Entry of "0" Entry is rejected with an error message.

This does not work, because "0" means "delete" and editors do not have the right to delete. The original entry "100" is retained.
10.2.5.2 Removing Server Entries for Users

You just saw how you as administrator can create a user with the "Palo Wizard". Of course, you can also delete the user. For example, when an employee leaves the company you delete his access by removing the server entry. Call the "Palo Wizard" to do this. Set "Test_Elke" as server.

Select "Unregister the selected server" entry under "Server Management". Confirm your selection by clicking "Finish".

Then the system asks you to confirm the deletion by clicking "OK".

Palo then reports: "Server successfully unregistered".
11 Working with Enterprise Rules

11.1 General Information on Rules

In previous Palo versions, the consolidation factor of elements was used to carry out simple calculations – this included additions and subtractions.

Palo Enterprise Rules enhance this functionality and enable you to execute complex calculations (division/multiplication, logical and various functions).

The general syntax of Rules is

\[ \text{Target} = f[\text{Source}] \]

i.e.\[ \text{Target} \] is a function of \[ \text{Source} \].

Whereas \[ \text{Target} \] is the area in the cube that is to be calculated using rules. This target area is defined by dimension elements. Elements are not to be listed, if a rule is to apply to all dimension elements.

Rules are mapped out per cube.

**IMPORTANT:** If a target area is defined in a rule, subsequent rules, which describe the same or an overlapping target area, are no longer effective. This means that specific rules have to be stored before general rules.

You can get to the Rule Editor via the context menu after a right-click on the cube in the Modeller, or alternatively, you can click on the symbol after you have selected a cube.
11.2 Creating a simple Rule

Simplified example using the Sales cube of the database "Demo": RULE 1

\[ ['2009'] = 123 \]

The target area is the year 2009 of the dimension "Years". This rule has the effect that all cells in the cube now contain the value 123 in the year 2009. The other years are not effected by this rule. Since for all other dimensions the elements were omitted, the rule applies in the other dimensions to all elements.

Open the Rule Editor:

For a new rule click on "NEW"
The Rule Editor (Advanced) opens:

For Target, select "Years" and "2009" and for Source enter the number "123". You can check whether the Rule Editor (Advanced) accepts your inputs, by clicking on the button "Parse". By clicking on OK, you confirm the rule.

Now create a view in order to check the result. Organize the view in such a way that the years 2006 up to 2009 become column headers. Place "Products" in the Row titles and select all products.
Some comments on this view:
The values in column E are retrieved from the corresponding cube-cell by the PALO.DATAC()-function. For 2009, this value was zero throughout. The values currently displayed in this view are computed using the rule, but the values stored in the data-base remain zero (i.e. the displayed values are not being written back to the data-base (Which is the case with entries made in cells!)).

The rule simply overwrites anything returned by PALO.DATAC() from the database. If you delete the rule, the corresponding original value from the database will be displayed again.
11.3 Restricting the Target Area

In the next step, please create a rule that will show for "Desktop L" in the year 2009 a value that is 10\% higher than in 2006.

The target area is "2009" and "Desktop L". The function for the source is "2006", "Desktop L" multiplied by 1.1.

If the target area contains a part of the source area, as in the example "Desktop L", it does not have to be repeated in the source area.

The conclusion is the following rule (RULE 2): \[['2009', 'Desktop L'] = ['2006']\*1.1

The target area is the year 2009 of the dimension "Years" of the product "Desktop L". This rule has the effect, that in 2009 all "Desktop L" cells of this cube show the 2006 value multiplied by 1.1. The other years and products respectively, are not affected by this rule.

IMPORTANT:
Since you already defined the target area for the entire year of 2009 in the previous rule (Rule 1), Rule 2 will only have an impact if you place it in front of the more general rule for 2009. You can change the order of execution using the arrows in the rule editor.

Besides a short notation, element names can be submitted along with dimension name.

Short form: \[['2009', 'Desktop L'] = ['2006']\*1.1

Long form: [[Years': '2009', 'Products': 'Desktop L'] = ['Years': '2006']\*1.1

This makes sense only when the same element name occurs under several dimensions, which is often the case with "Total".

By clicking on the star (red arrow) more address-fields become visible.
You can select the operator with the button "...".

Select the asterisk for multiplication.

Enter 1.1 as factor and then click on OK.

In order to get the correct ordering, the new rule must be placed before the rule \[2009\] = 123.
Now please verify the result in your view:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>localhost/Demo</strong></td>
<td><strong>Sales</strong></td>
<td><strong>Europe</strong></td>
<td><strong>Year</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Units</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Budget</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2006</td>
<td>2007</td>
<td>2008</td>
<td>2009</td>
<td></td>
</tr>
<tr>
<td>9.651.627,31</td>
<td>10.616.790,04</td>
<td>0,00</td>
<td>123,00</td>
<td></td>
</tr>
<tr>
<td>3.741.322,51</td>
<td>4.115.455,86</td>
<td>0,00</td>
<td>123,00</td>
<td></td>
</tr>
<tr>
<td>636.705,43</td>
<td>700.375,97</td>
<td>0,00</td>
<td>700.375,97</td>
<td></td>
</tr>
<tr>
<td>608.765,56</td>
<td>656.642,11</td>
<td>0,00</td>
<td>123,00</td>
<td></td>
</tr>
<tr>
<td>488.887,23</td>
<td>535.575,95</td>
<td>0,00</td>
<td>123,00</td>
<td></td>
</tr>
<tr>
<td>307.596,55</td>
<td>338.356,31</td>
<td>0,00</td>
<td>123,00</td>
<td></td>
</tr>
<tr>
<td>233.066,86</td>
<td>256.373,55</td>
<td>0,00</td>
<td>123,00</td>
<td></td>
</tr>
<tr>
<td>376.116,17</td>
<td>413.727,79</td>
<td>0,00</td>
<td>123,00</td>
<td></td>
</tr>
<tr>
<td>270.243,06</td>
<td>306.068,25</td>
<td>0,00</td>
<td>123,00</td>
<td></td>
</tr>
<tr>
<td>319.947,72</td>
<td>351.942,49</td>
<td>0,00</td>
<td>123,00</td>
<td></td>
</tr>
<tr>
<td>239.705,78</td>
<td>263.676,35</td>
<td>0,00</td>
<td>123,00</td>
<td></td>
</tr>
<tr>
<td>254.283,25</td>
<td>279.717,08</td>
<td>0,00</td>
<td>123,00</td>
<td></td>
</tr>
</tbody>
</table>

If the rule ['2009'] = 123 would be on top, previous rule-results would have been overwritten with 123. So, however, the value 123 is overwritten with the value 636705,43 * 1.1 = 700375,97 in cell E11. Possible prior consolidations will be consistently overwritten by the rules without any error message (cells E9 and E10).

### 11.4 Operators of the Rule-Editor

You can select the operator with the button "...".

You will then receive the following selection:

The operators which are not yet known from Excel will now be described:

- == is equal to (in Excel = is used)
- != is not equal (in Excel <> is used)
- These operators may be used in the IF function.
- @ is used only for markers (this is described in chapter "Markers for Expressions")
- Remove is the switch to delete the operator on the left side of the "..." button.

This handbook has been personalized for NA - ohneanmeldung
11.5  Restricting the Target Area to an Element Type

If a rule is to apply to base elements only (consolidated elements will be aggregated as usual), the target area may be defined as follows:

\[ \text{[Target]} = N: f[\text{Source}] \]

Vice versa, it is also possible to restrict the target area to consolidated elements.

\[ \text{[Target]} = C: f[\text{Source}] \]

Please change both rules as follows and recalculate, i.e. refresh your table:

You can also achieve this modification in the Rule Editor (Advanced) by selecting "Base Elements" in the list box "Consolidation priority" (red arrow).

\[['2009'] = N:123
\[['2009','Desktop L'] = N: ['2006'] * 1.1

Now compute the table anew.

Since the target area is now reduced to base elements, the consolidated elements will be aggregated as usual.
11.6 Drill Palo Rule

The command "Drill Palo Rule" is available with right-click on the cell E9. So you get the following Rule Info:

```
Rule Info

activated: True

comment

definition: ['2009','Desktop L'] = N['2006'] * 1.1

event_id: 1

timestamp: 1228138407
```

If you enter this command on a cell without a rule, then you get the following message:
11.7 Accessing a Different Cube Via Rule

For the following example, please create a cube with the name "Sales2" (same dimensions and order of dimensions as in "Sales").

Now please create the following rule for the "Sales2" cube:

The 'Budget'-values for 2008 of the cube Sales2 should be twice as large as the 'Actual'-values for 2006 of the cube Sales:

$$[\text{2008}',\text{Budget}'] = \text{PALO.DATA}('\text{Demo}', '\text{Sales}', '\text{Products}', '\text{Regions}', '\text{Months}', '2006', 'Actual', '\text{Measures}') \times 2$$

The Rule Editor (Advanced) assists us here also, if we select PALO.DATA as Source:

To check it, create the following view:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo</td>
<td></td>
<td></td>
<td>localhost/Demo</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sales2</td>
<td></td>
<td></td>
<td>Sales</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Europe</td>
<td></td>
<td></td>
<td>Europe</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Year</td>
<td></td>
<td></td>
<td>Year</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Units</td>
<td></td>
<td></td>
<td>Units</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Budget</td>
<td></td>
<td></td>
<td>Actual</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>2008</td>
<td>2009</td>
<td></td>
<td>2006</td>
</tr>
<tr>
<td>8</td>
<td>All Products</td>
<td>20,200,689.34</td>
<td>0.00</td>
<td>10,100,349.17</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Stationary PC's</td>
<td>7,784,948.42</td>
<td>0.00</td>
<td>3,882,474.21</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Desktop L</td>
<td>1,350,101.31</td>
<td>0.00</td>
<td>675,050.66</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Desktop Pro</td>
<td>1,299,040.53</td>
<td>0.00</td>
<td>649,520.27</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Desktop Pro XL</td>
<td>1,014,009.43</td>
<td>0.00</td>
<td>507,403.21</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Desktop High XL</td>
<td>644,268.59</td>
<td>0.00</td>
<td>322,134.30</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Desktop High XG</td>
<td>473,620.53</td>
<td>0.00</td>
<td>236,814.27</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Server Power XC</td>
<td>747,973.45</td>
<td>0.00</td>
<td>373,939.73</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Server Power TT</td>
<td>966,857.89</td>
<td>0.00</td>
<td>293,420.85</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Server Dual C</td>
<td>626,919.07</td>
<td>0.00</td>
<td>313,459.54</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Server Dual XC</td>
<td>534,302.86</td>
<td>0.00</td>
<td>257,151.43</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Server Lion RX</td>
<td>527,043.96</td>
<td>0.00</td>
<td>253,521.98</td>
<td></td>
</tr>
</tbody>
</table>
11.8 Usage of Markers

Markers are labels in rules, set in order to process performance-critical rules preferentially. In Palo, markers are defined within the rule.

Markers can only be defined for N:Rules respectively in rules with markers are only computed the base elements.

11.8.1 Advantages of Markers

By using markers, performance-critical rules will be processed preferentially. The decision to use markers is not a global one, which applies to the entire cube, but individual rules can get markers to accelerate their calculation.

Evaluation of an aggregation is performed as follows:
If the sub-cube spanned by the aggregation contains no rules or only rules without marker, the standard algorithm will be used. If, however, the sub-cube contains at least one rule with a marker, the marker algorithm will be used and rules without a marker will be ignored. The advantage of this procedure is, that only performance-critical rules need to be defined with markers.

Important Note:
Accelerated calculation of marker-Rules is achieved in these Rules shall be calculated based only on those cells whose value is not 0 (zero). All the basic cells with a value of 0 will be ignored. The result is a massive acceleration, but also the following restriction:
For correct results, only those rules may be provided with markers, whose result is always 0 if all marked arguments have the value 0.

11.8.2 Disadvantages of Markers

If markers are used, the initial loading of the databases can be slowed down.
If too many markers are used, too much memory will be consumed, thereby destroying the performance gain (so called over marking).
Creating, modifying and deleting rules with markers takes more time than it would take without markers.

11.8.3 How are Markers defined?

Ordinarily, a N:Rule is usually defined as follows:

\[
[\text{Turnover}] = N: [\text{Quantity}] \times [\text{Price}]
\]

Now, if one of the source-areas involved shall be labeled with a marker, the pair of double-braces "[[' ']]" is used instead of the pair "[' ']". If "Quantity" should get a marker, one writes:

\[
[\text{Turnover}] = N: [[\text{Quantity}]] \times [\text{Price}]
\]

This defines a marker from ['Quantity] to ['Turnover']. This means that the value ['turnover'] is calculated only for those basic cells, where the value of ['quantity'] is not 0. (This makes sense, since with a quantity of 0 the turnover will also be 0).

It can happen, that more dimensions are specified in the marker than in the target area:

\[
[\text{2005}', \text{Turnover}] = N: [[\text{2003'}, \text{Turnover'}, \text{Cola}']] + [[\text{2004'}, \text{Turnover}']
\]

In this case, the marker will be restricted correspondingly, i.e. from [2003', 'Turnover', 'Cola'] to [2005', 'Turnover', 'Cola'].

It is possible to define several markers:

\[
[\text{2005}', \text{Turnover}] = N: [[\text{2003'}, \text{Turnover'}] + [[\text{2004'}, \text{Turnover'}
\]

In this case, two markers are defined. One marker from [2003', 'Turnover'] to [2005', 'Turnover'] and one marker from [2004', 'Turnover'] to [2005', 'Turnover']. In this case, the rule is calculated only if at least one of the marked arguments on the right side is different from 0.
It is also possible to define markers for entire cubes. The simplest case is the case in which both cubes have the same dimensions:

\[
[\text{'2005','Turnover'}] = N: \text{PALO.MARKER("Database","Planning-cube",'2004','Turnover')}
\]

In this case, a marker from ['2004','Turnover'] in the Planning-cube to ['2005','Turnover'] is defined. In case that the marker-cube has additional dimensions, these have to be specified with constants. It follows an example with the name of the planner, Mr. Miller:

\[
[\text{'2005','Turnover'}] = N: \text{PALO.MARKER("Database","Planning-cube","Mr. Miller','2004','Turnover')}
\]

It defines the marker from ['Mr. Miller', '2004', 'Turnover'] in the Planning-cube to ['2005','Turnover'].

If the marker-cube has fewer dimensions than the target cube, the target area can be specified without any restrictions or it can be restricted by specifying a base element:

\[
[\text{'Turnover'}] = N: \text{PALO.MARKER("Database","Planning-cube","Mr. Miller','Turnover')}
\]

or

\[
[\text{'2005','Turnover'}] = N: \text{PALO.MARKER("Database","Planning-cube","Mr. Miller','Turnover')}
\]

### 11.8.4 Arguments of PALO.MARKER()

Only constants and variables can be used as arguments for PALO.MARKER(). General expressions are not allowed.

Example:

\[
[\text{'2005','Turnover'}] = N: \text{IF (['Price']>10, PALO.MARKER(…)*2, PALO.MARKER(…)*3)}
\]

This is allowed, because the IF-expression does not occur in the argument of the marker.

\[
\]

This is not allowed, because the IF-expression does occur in the argument of the marker.

### 11.8.5 Markers for Expressions

If you wish to mark expressions (Formulas/Functions), you can, for example, append "@" PALO.MARKER() to PALO.DATA().

The following rule gives an example:

\[
[\text{'2005','Turnover'}] = N: \text{PALO.DATA("Database","Planning-cube",MID(!'Year',1,4),'Turnover')}
\]

Here you can define a marker as follows:

\[
[\text{'2005','Turnover'}] = N: \text{PALO.DATA("Database","Planning-cube",MID(!'Year',1,4),'Turnover')} @ \text{PALO.MARKER("Database","Planning-cube", 'AllYears%', 'Turnover')}
\]

In this case, a marker from [all children of 'AllYears%', 'Turnover'] of the planning-cube to ['2005','Turnover'] is defined.
11.8.6 Processing of the Rules

In case that more than one rule is defined in a cube, the rules will be processed as follows:

Is the cell to be calculated a base cell, the first this cell matching rule is used. Its result is the value of the base cell.

Is the cell to be calculated a consolidated cell and exists a matching rule, which isn't a N-Rule, this rule will be used. Its result is the value of the consolidated cell.

If a consolidated cell is concerned and if there exists a base cell in the set of base cells of the consolidation, for which an N:Rule with marker fits, the marker-consolidation-algorithm will be used. In this case, only cells which have been marked with a marker will be used for consolidation. However, the first fitting rule will be used as value of the marked base cells, irrespective of the presence of markers in that rule.

In all other cases, the standard algorithm is used.

11.9 Examples of Rules Using Demo Database Demo2

In the following section, we will use examples to describe a few rules for the Demo2 database. You can download the database along with additional information on the data model and also some test views with regards to below rules at http://www.jedox.com/en/products/palo_olap_server/download.html.

11.9.1 Example Rule 1

The following rule is to calculate the variance between act and bud in per cent. Target area is the element "variance\%" in dimension "opstmt_type". The rule applies to both base elements and also consolidated elements.

\[
['variance\%'] = (['variance'] / ['bud']) * 100
\]

Below a test view with the results of Rule 1 in column E:

<table>
<thead>
<tr>
<th></th>
<th>act</th>
<th>bud</th>
<th>variance</th>
<th>variance%</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Localhost Demo2</td>
<td>7,765,1</td>
<td>7,083,46</td>
<td>676,53</td>
</tr>
<tr>
<td>2</td>
<td>OPSTMT</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>3</td>
<td>localcur</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>4</td>
<td>2007</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>5</td>
<td>37</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>6</td>
<td>General administration</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>7</td>
<td>Other income/loss</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>8</td>
<td>gernany</td>
<td>7,255,91</td>
<td>607,83</td>
<td>118,08</td>
</tr>
<tr>
<td>9</td>
<td>National Research Technologies</td>
<td>7,568</td>
<td>6,23</td>
<td>1,35</td>
</tr>
<tr>
<td>10</td>
<td>General Research Company</td>
<td>104,23</td>
<td>85,24</td>
<td>19,00</td>
</tr>
<tr>
<td>11</td>
<td>Eastern Products Group</td>
<td>626,74</td>
<td>301,91</td>
<td>26,53</td>
</tr>
<tr>
<td>12</td>
<td>Eastern Energy Office</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>13</td>
<td>Multinational Consulting, Inc.</td>
<td>7,255,91</td>
<td>607,83</td>
<td>118,08</td>
</tr>
<tr>
<td>14</td>
<td>Western Chemicals Technologies</td>
<td>7,568</td>
<td>6,23</td>
<td>1,35</td>
</tr>
<tr>
<td>15</td>
<td>Northern Chemicals Agency</td>
<td>104,23</td>
<td>85,24</td>
<td>19,00</td>
</tr>
<tr>
<td>16</td>
<td>Northern Iron Co. Ltd.</td>
<td>626,74</td>
<td>301,91</td>
<td>26,53</td>
</tr>
<tr>
<td>17</td>
<td>Northern Iron Holding</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>18</td>
<td>Professional Investments Office</td>
<td>100,00</td>
<td>0,00</td>
<td>100,00</td>
</tr>
<tr>
<td>19</td>
<td>General Chemicals Company</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>20</td>
<td>General Energy Co. Ltd.</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>21</td>
<td>Professional Chemicals Technologies</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>22</td>
<td>Northern Electronics Office</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
</tbody>
</table>
11.9.2 Example Rule 2

Rule 2 puts the P&L Report positions into a percentile ratio to the turnover (Position "A00140"). The element "as%ofSales" is included in dimension "opstmt_curr".

\[ ['as\%ofSales'] = ('localcurr') / ('A00140','localcurr') \times 100 \]

Below a view for testing purposes with the result of Rule 2 in column C:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>OPSTMT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>act</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2007</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>europa</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>total</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>Here you can see values as percentage of net sales</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>localcurr</td>
<td>as%ofSales</td>
</tr>
<tr>
<td>10</td>
<td>net sales</td>
<td>23,035,518,59</td>
<td>100,00</td>
</tr>
<tr>
<td>11</td>
<td>calculated sales</td>
<td>24,510,108,74</td>
<td>102,93</td>
</tr>
<tr>
<td>12</td>
<td>reduction in earnings 2</td>
<td>674,690,16</td>
<td>2,83</td>
</tr>
<tr>
<td>13</td>
<td>imputed interest on receivables/credits</td>
<td>404,754,09</td>
<td>1,70</td>
</tr>
<tr>
<td>14</td>
<td>imputed interest on receivables 1</td>
<td>89,948,35</td>
<td>0,38</td>
</tr>
<tr>
<td>15</td>
<td>imputed interest on receivables 2</td>
<td>44,972,88</td>
<td>0,19</td>
</tr>
<tr>
<td>16</td>
<td>income from loan</td>
<td>134,918,03</td>
<td>0,57</td>
</tr>
<tr>
<td>17</td>
<td>imputed Interest on loan</td>
<td>134,918,03</td>
<td>0,57</td>
</tr>
<tr>
<td>18</td>
<td>reduction in earnings 1</td>
<td>259,836,06</td>
<td>1,13</td>
</tr>
</tbody>
</table>

This handbook has been personalized for NA - ohneanmeldung
11.9.3 Example Rule 3

Rule 3 is somewhat more complicated, as it contains nested PALO.DATA formulas. The rule converts "localcurr" values into Euros (on the basis of exchange rates stored in the "currency" cube). You will find sample exchange rates in the month "06" of the year 2007. The currency applicable to the corporation is stored in the attribute "local currency" of the dimension "opstmt_entity".

The PALO.DATA formulas are represented in different colors:

\[
['EUR'] = N: ['localcurr'] * PALO.DATA("Demo2","CURRENCY",
PALO.DATA("Demo2","#_opstmt_entity","local currency","!'opstmt_entity'),
!'opstmt_year',!'opstmt_month')
\]

The outer PALO.DATA-formula is retrieving the exchange rate from the cube "currency", whereas one of the parameters is defined by another PALO.DATA formula to obtain the currency sign contained in the attribute cube "#_opstmt_entity".

Here the results of the exchange rate calculation in column C:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo2</td>
<td>OPSTMT</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>act</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2007</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>balance sheet profit/loss</td>
<td>EUR is calculated by rule</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>localcurr</td>
<td>EUR</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>world</td>
<td>6,523,266,93</td>
<td>406,607,727,66</td>
</tr>
<tr>
<td>11</td>
<td>europe</td>
<td>3,034,212,86</td>
<td>18,199,982,95</td>
</tr>
<tr>
<td>12</td>
<td>algeria</td>
<td>56,975,71</td>
<td>5,487,624,20</td>
</tr>
<tr>
<td>13</td>
<td>Eastern Electronics Co. Ltd.</td>
<td>48,725,22</td>
<td>49,212,47</td>
</tr>
<tr>
<td>14</td>
<td>belgium</td>
<td>44,967,39</td>
<td>45,417,06</td>
</tr>
<tr>
<td>15</td>
<td>Western Energy Group</td>
<td>3,757,83</td>
<td>3,795,41</td>
</tr>
<tr>
<td>16</td>
<td>General Cash Holding</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>17</td>
<td>Multinational Research Co. Ltd.</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>18</td>
<td>General Research Co. Ltd.</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>19</td>
<td>General Trading Corporation</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>20</td>
<td>International Investments Technologies</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>21</td>
<td>denmark</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>22</td>
<td>Northerm Research Technologies</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>23</td>
<td>Western Iron , inc.</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>24</td>
<td>National Tracing Ltd.</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>25</td>
<td>finland</td>
<td>8,016,62</td>
<td>6,096,79</td>
</tr>
</tbody>
</table>
11.9.4 Example Rule 4

The 4th rule fills the elements with values depicting the variation of 2 years in percentage – example VAR_08-07% - in the dimension "opstmt_year".

The rule for the variation VAR_08-07% as follows:

\[
['VAR_08-07\%'] = \left(\frac{['VAR_08-07','localcurr']}{['2008','localcurr']}\right) \times 100
\]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>localhostDemo2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPSTMT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>localcurr</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>act</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>07</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>general administration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>other incomes/loss</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gernany</td>
<td>5,961,18</td>
<td>7,765,11</td>
<td>13,726,29</td>
<td>43.43</td>
</tr>
<tr>
<td>National Research Technologies</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>General Research Company</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Eastern Products Group</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Eastern Energy Office</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Multinational Consulting, Inc.</td>
<td>716.69</td>
<td>725.21</td>
<td>1,442.00</td>
<td>49.66</td>
</tr>
<tr>
<td>Western Chemicals Technologies</td>
<td>0,76</td>
<td>7,60</td>
<td>8,33</td>
<td>9.03</td>
</tr>
<tr>
<td>Northam Chemicals Agency</td>
<td>10,42</td>
<td>104,23</td>
<td>114,86</td>
<td>9.08</td>
</tr>
<tr>
<td>Northam Iron Co. Ltd</td>
<td>151.38</td>
<td>520,74</td>
<td>900.42</td>
<td>15.47</td>
</tr>
<tr>
<td>Northam Iron Holding</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Professional Investments Office</td>
<td>100,00</td>
<td>100,00</td>
<td>200,00</td>
<td>50.00</td>
</tr>
<tr>
<td>General Chemicals Company</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>General Energy Co. Ltd.</td>
<td>100,00</td>
<td>0,00</td>
<td>100,00</td>
<td>100.00</td>
</tr>
<tr>
<td>Professional Chemicals Technologies</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>Northam Electronics Office</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
<td>0,00</td>
</tr>
<tr>
<td>General Research Technologies</td>
<td>160,98</td>
<td>464,12</td>
<td>625,00</td>
<td>25.74</td>
</tr>
</tbody>
</table>

This handbook has been personalized for NA - ohneanmeldung
11.10 Supported Functions for Rules

Functions supported by the Server are listed under the API-Functions. Please open your browser and enter "localhost:7777/api". The functions are listed in the API documentation under /rule/functions.

Palo Server Rules have a time system with 1.1.1970 as start time and have seconds as unit. e.g.: 1.1.1970 00:00:00 = 0, 2.1.1970 00:00:00 = 86400 (= 24h * 60min * 60sec).

Please find below a summary of the functions with a brief description:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS</td>
<td>Returns the absolute value of a number. The absolute value of a number is the number without its sign.</td>
</tr>
<tr>
<td>ACOS</td>
<td>Inverse cosine function. Return value in radians.</td>
</tr>
<tr>
<td>ADD</td>
<td>Returns the sum of the two arguments.</td>
</tr>
<tr>
<td>AND</td>
<td>AND(Expression1, expression2). If both expressions are true, then TRUE, otherwise FALSE is returned.</td>
</tr>
<tr>
<td>ASIN</td>
<td>Inverse sine function. Return value in radians.</td>
</tr>
<tr>
<td>ATAN</td>
<td>Inverse tangents function. Return value in radians.</td>
</tr>
<tr>
<td>AVERAGE</td>
<td>Returns the average of its arguments.</td>
</tr>
<tr>
<td>CEILING</td>
<td>Rounds a number to the nearest integer or to the nearest multiple of significance.</td>
</tr>
<tr>
<td>CHAR</td>
<td>Returns the character specified by a number.</td>
</tr>
<tr>
<td>CLEAN</td>
<td>Removes all nonprintable characters from a string.</td>
</tr>
<tr>
<td>CODE</td>
<td>Returns the code number of the first character in the string argument.</td>
</tr>
<tr>
<td>CONCATENATE</td>
<td>Joins 2 strings together.</td>
</tr>
<tr>
<td>CONTINUE</td>
<td>If a rule evaluates to CONTINUE then that rule is skipped and search for valid rule is continued.</td>
</tr>
<tr>
<td>COS</td>
<td>Returns the cosine of the argument which is in radians.</td>
</tr>
<tr>
<td>COUNT</td>
<td>Returns the number of values provided.</td>
</tr>
<tr>
<td>DATE</td>
<td>Returns the serial number of a particular date.</td>
</tr>
<tr>
<td>DATEFORMAT</td>
<td>Converts date (serial number) to date in text format.</td>
</tr>
<tr>
<td>DATEVALUE</td>
<td>Converts a string to a date.</td>
</tr>
<tr>
<td>DEL</td>
<td>Returns the second argument subtracted from the first argument</td>
</tr>
<tr>
<td>DIV</td>
<td>Returns the quotient and remainder of the first value divided by the second</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>EQ</td>
<td>Checks if the two arguments, which can be of type string or double, are exactly equal.</td>
</tr>
<tr>
<td>EVEN</td>
<td>Returns number rounded up to the nearest even integer</td>
</tr>
<tr>
<td>EXACT</td>
<td>Compares two text strings and returns TRUE if they are exactly the same. FALSE otherwise. Case-Sensitive</td>
</tr>
<tr>
<td>EXP</td>
<td>Returns e raised to the power of number applied.</td>
</tr>
<tr>
<td>FACT</td>
<td>Returns the factorial of a number</td>
</tr>
<tr>
<td>FIRST</td>
<td>Returns first of its arguments.</td>
</tr>
<tr>
<td>FLOOR</td>
<td>Rounds number down, toward zero</td>
</tr>
<tr>
<td>GE</td>
<td>Checks if first argument is greater than or equal to the second. Both arguments must be of the same type which can be double or string.</td>
</tr>
<tr>
<td>GT</td>
<td>Checks if first argument is greater than the second. Both arguments must be of the same type which can be double or string.</td>
</tr>
<tr>
<td>IF</td>
<td>IF checks, if the first parameter is TRUE. In this case, the second parameter is returned. Otherwise, the third parameter is returned.</td>
</tr>
<tr>
<td>INT</td>
<td>Rounds a number down to the nearest integer.</td>
</tr>
<tr>
<td>ISERROR</td>
<td>ISERROR returns TRUE if the expression cannot be evaluated or the evaluation isn't a string or a numeric value.</td>
</tr>
<tr>
<td>LAST</td>
<td>Returns last of its arguments.</td>
</tr>
<tr>
<td>LE</td>
<td>Checks if first argument is less than or equal to the second. Both arguments must be of the same type which can be double or string.</td>
</tr>
<tr>
<td>LEFT</td>
<td>Returns the first character or characters in a text string, based on the number of characters specified.</td>
</tr>
<tr>
<td>LEN</td>
<td>Returns the number of characters in a text string.</td>
</tr>
<tr>
<td>LN</td>
<td>Returns the natural logarithm of the integer argument.</td>
</tr>
<tr>
<td>LOG</td>
<td>Returns the logarithm of a number to the base specified.</td>
</tr>
<tr>
<td>LOG10</td>
<td>Returns the base-10 logarithm of the argument.</td>
</tr>
<tr>
<td>LOWER</td>
<td>Converts all uppercase letters in a text string to lowercase.</td>
</tr>
<tr>
<td>LT</td>
<td>Checks if first argument is less than the second. Both arguments must be of the same type which can be double or string.</td>
</tr>
<tr>
<td>MAX</td>
<td>Returns the maximum value of its arguments.</td>
</tr>
<tr>
<td>MID</td>
<td>Returns a specific number of characters from a text string, starting at the position you specify, based on the number of characters you specify.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MIN</td>
<td>Returns the minimum value of its arguments.</td>
</tr>
<tr>
<td>MOD</td>
<td>Returns the remainder after number is divided by divisor. The result has the same sign as divisor.</td>
</tr>
<tr>
<td>MUL</td>
<td>Returns the arithmetic product of the two arguments.</td>
</tr>
<tr>
<td>NE</td>
<td>Checks if first argument is equal to the second. Returns True if not equal.</td>
</tr>
<tr>
<td>NOT</td>
<td>NOT(Boolean expression): Returns the logical inversion of the Boolean expression back.</td>
</tr>
<tr>
<td>NOW</td>
<td>Returns the serial number of the current date and time.</td>
</tr>
<tr>
<td>ODD</td>
<td>Returns number rounded up to the nearest odd integer.</td>
</tr>
<tr>
<td>OR</td>
<td>OR(Expression1, expression2). If at least one expression is true, then TRUE, otherwise FALSE is returned.</td>
</tr>
<tr>
<td>PALO.CUBEDIMENSION</td>
<td>Retrieves the name of the ( n )th dimension in a specified hypercube.</td>
</tr>
<tr>
<td>PALO.DATA</td>
<td>Retrieves the value of the specified element from the cube</td>
</tr>
<tr>
<td>PALO.ECHILD</td>
<td>Retrieves the name of the specified child element</td>
</tr>
<tr>
<td>PALO.ECHILDCOUNT</td>
<td>Retrieves the number of children in the specified consolidated element</td>
</tr>
<tr>
<td>PALO.ECOUNT</td>
<td>Retrieves the amount of dimension elements in the specified dimension</td>
</tr>
<tr>
<td>PALO.EFIRST</td>
<td>Retrieves the first element in the specified child dimension</td>
</tr>
<tr>
<td>PALO.EINDENT</td>
<td>Retrieves the indentation level of the specified element</td>
</tr>
<tr>
<td>PALO.EINDEX</td>
<td>Retrieves the position of the specified dimension element</td>
</tr>
<tr>
<td>PALO.EISCHILD</td>
<td>Checks, whether a consolidated element contains a specified base element. It returns 0 or 1.</td>
</tr>
<tr>
<td>PALO.ELEVEL</td>
<td>Checks if a consolidated element contains the specified element, results in 0 or 1</td>
</tr>
<tr>
<td>PALO.ENAME</td>
<td>Retrieves the name of the element at the specified position (First Position is 1)</td>
</tr>
<tr>
<td>PALO.ENEXT</td>
<td>Retrieves the name of the succeeding element of a dimension element</td>
</tr>
<tr>
<td>PALO.EPARENT</td>
<td>Retrieves the name of the ( n )th parent of the specified element</td>
</tr>
<tr>
<td>PALO.EPARENTCOUNT</td>
<td>Retrieves the number of consolidated elements which contain the specified element.</td>
</tr>
<tr>
<td>PALO.EPREV</td>
<td>Retrieves the name of the preceding element of a dimension element</td>
</tr>
<tr>
<td>PALO.ESIBLING</td>
<td>Retrieves the name of the specified sibling</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PALO.ETOLEVEL</td>
<td>Returns the level number of the highest element in the consolidation hierarchy of a dimension.</td>
</tr>
<tr>
<td>PALO.ETYPE</td>
<td>Retrieves the type of the specified element (numeric, string or consolidated)</td>
</tr>
<tr>
<td>PALO.EWEIGHT</td>
<td>Returns the consolidation weight of a specified component of an element.</td>
</tr>
<tr>
<td>PALO.MARKER</td>
<td>Adds marker for a slice.</td>
</tr>
<tr>
<td>PI</td>
<td>Returns the value of the constant pi</td>
</tr>
<tr>
<td>POWER</td>
<td>Returns the result of a number raised to a power</td>
</tr>
<tr>
<td>PROPER</td>
<td>Capitalizes the first letter in a text string and any other letters in text that follow any character other than a letter. Converts all other letters to lowercase</td>
</tr>
<tr>
<td>QUOTIENT</td>
<td>Returns the integer portion of a division</td>
</tr>
<tr>
<td>RAND</td>
<td>Returns a random number between 0 and 1</td>
</tr>
<tr>
<td>RANDBETWEEN</td>
<td>Returns a random number between the numbers you specify</td>
</tr>
<tr>
<td>REPLACE</td>
<td>Replaces part of a text string, based on the number of characters specified, with a different text string</td>
</tr>
<tr>
<td>REPT</td>
<td>Repeats text a given number of times</td>
</tr>
<tr>
<td>RIGHT</td>
<td>Returns the last character or characters in a text string, based on the number of characters specified</td>
</tr>
<tr>
<td>ROUND</td>
<td>Rounds a number to a specified number of digits.</td>
</tr>
<tr>
<td>SEARCH</td>
<td>Finds one text value within another (not case-sensitive). Find_text is the text you want to find, e.g. SEARCH(&quot;x&quot;,&quot;Text&quot;) returns 1 (TRUE).</td>
</tr>
<tr>
<td>SIGN</td>
<td>Returns the sign of a number.</td>
</tr>
<tr>
<td>SIN</td>
<td>Returns the sine of the given angle</td>
</tr>
<tr>
<td>SQRT</td>
<td>Returns a positive square root</td>
</tr>
<tr>
<td>STET</td>
<td>if a rule evaluates to STET, then the resulting cell behave like no rule is existent for this cell</td>
</tr>
<tr>
<td>STR</td>
<td>Converts number to string.</td>
</tr>
<tr>
<td>SUBSTITUTE</td>
<td>Substitutes new_text for old_text in a text string.</td>
</tr>
<tr>
<td>SUM</td>
<td>Returns sum of values provided through function arguments.</td>
</tr>
<tr>
<td>TAN</td>
<td>Returns the tangent of the given angle</td>
</tr>
<tr>
<td>TRIM</td>
<td>Removes spaces from text</td>
</tr>
</tbody>
</table>
TRUNC | Truncates a number to an integer by removing the fractional part of the number
---|---
UPPER | Converts text to uppercase
VALUE | Converts a text string to a number. Returns 0 if conversion fails.
VALUEDATE | Return a date_text for a given serial number in the format MM-TT-YY
WEEKDAY | Returns the day of the week corresponding to a date. The day is given as an integer, ranging from 1 (Sunday) to 7 (Saturday), by default.

You can find details on the individual functions in the Rules section of the API.
12 Filter Functions with Subsets

A subset is a set within a dimension. With a subset you can select elements or sets of elements and sort them according to certain criteria. Example: you would like to get the Top 10 items of a list of products and you would like for the results to be within defined marginal values.

The function PALO.SUBSET() is the basic filter, which all the other filters require to display data.

Filters can have an impact on a subset in three different ways:
- Restricting by removing elements
- Structuring by changing the order and possibly the frequency of the elements or by directly adding elements
- Indirectly by changing the behavior of other filters

12.1 PALO.SUBSET()

The function PALO.SUBSET() has the following arguments:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server/Database</td>
<td>Text</td>
<td>e.g. localhost/Demo or A1 (the cell which contains the entry)</td>
</tr>
<tr>
<td>Dimension</td>
<td>Text</td>
<td>e.g. Products or A2 (the cell which contains the name of the dimension).</td>
</tr>
<tr>
<td>Indent</td>
<td>Number</td>
<td>The allowed values are 1 (equivalently: empty), 2 and 3. The argument alters the numbering of the hierarchy-level.</td>
</tr>
<tr>
<td>Alias</td>
<td>Text / Array</td>
<td>Contains at most two aliases. {&quot;Alias1&quot;,&quot;Alias2&quot;} or &quot;Alias&quot;. Alternatively, it is allowed to use a reference to the cell, e.g. {A1} or {A1:A2}. If empty, aliases are deactivated.</td>
</tr>
<tr>
<td>Filter</td>
<td>Filter name()</td>
<td>Allowed are: PALO.HFILTER(), PALO.TFILTER(), PALO.AFILTER(), PALO.DFILTER(), PALO.PICKLIST() und PALO.SORT().</td>
</tr>
</tbody>
</table>

12.1.1 Indent

The use of "Indent" alters the numbering of the hierarchy-level of the elements. This affects the subset, where the hierarchy-level is specified in the filter (see chapter: Details of PALO.SUBSET() ).

12.1.2 Alias

The use of "Alias" has the effect, that the sorting-filters do not use the original names of the elements for their operations, but that they use the aliases defined in the attribute-cube instead.

12.1.3 Filters and their Effects

All the other filters are sub-functions of PALO.SUBSET(), which means, that they cannot work independently but require PALO.SUBSET() as base filter.
12.1.3.1 Hierarchy-Filter PALO.HFILTER()
This filter is both structuring as well as restricting. It is possible to define the restrictions of a subset via the hierarchy within the dimension (e.g. all elements below a specific element) (restricting feature). Further, it is here that you can define for the subset to revolve. This means that the elements are being repeated until a defined number of elements has been reached (structuring).

12.1.3.2 Text-Filter PALO.TFILTER()
The text filter is restricting. Elements which fit a particular pattern, will be included, the rest will be eliminated from being displayed.

12.1.3.3 Pick list-Filter PALO.PICKLIST()
With the pick list-filter one can
- either add elements directly, so that they cannot be removed from the subset by any means
- or make a pre-selection of elements. The remaining filters of the subset will only be applied to the selected elements.

12.1.3.4 Attribute-Filter PALO.AFILTER()
This filter has a restrictive impact. Element attributes may be used as selective criteria. It differs from Alias, which also draws on attribute elements, insofar as the Alias has an indirect impact in terms of sorting (e.g. in connection with the text filter). The attribute filter, however, is an independent filter that has a restrictive impact. You can use this filter e.g. to eliminate elements from a selection if they don’t match a particular attribute.

12.1.3.5 Data-Filter PALO.DFILTER()
The data filter acts restrictive and indirectly. For each element a data value is calculated. If the data value meets the criteria, the element will be kept, otherwise it will be dismissed (restricting). And further, it can be sorted by the calculated data values using the sort filter (indirect).

12.1.3.6 Sorting-Filter PALO.SORT()
With the sorting-filter one can determine the sequence of the elements in the subset (e.g. sorting by data-value, alphabetically or by alias). If the sorting-filter is not used, the elements will be displayed in the order of creation.

12.1.4 Sequence of Execution
The order in which you enter the filters, is not relevant, however, Palo will execute the filters in a pre-defined sequence.

In the first instance, all indirectly operating filters are activated. With filters that have several characteristics, e.g. indirect and restricting, only the indirect part will be active at first. All indirect characteristics act simultaneously without affecting each other.

Then the restricting filters will be applied. If the "Preselecting" argument is chosen in the pick list-filter, only the selected elements are passed on to the further filtering of the subset, otherwise all elements of the chosen dimension are passed on. With filters that have several characteristics, e.g. restricting and structuring, only the restricting part will be executed. The order of execution is irrelevant, because elements get merely removed from the selection.
Finally, the structuring filters are applied. Here, there is an exactly defined sequence that results inevitably from the filters:

- At first, there is a list of those elements which have not been eliminated by the previous filters. The sequence is the sequence in which they are defined in the Modeller.
- If the argument "Insert" has been chosen in the PALO.PICKLIST()-function, its elements will inserted in the order of creation.
- If PALO.SORT() has been specified with arguments, then its instructions will be executed now.
- If the argument "Insert before the list" or "Insert after the list" has been chosen in the PALO.PICKLIST()-function, this instruction will be executed afterwards.
- Finally, the revolving-arguments of the hierarchical filter will be executed.

### 12.2 Subset Editor

As of PALO-version 2.5, there is the Subset Editor. It makes it much easier to enter a subset-function than it was before, with manually defined subset-functions:

In the individual register-cards, you can simply click on the arguments or enter expressions for the individual filters. In the dialog-window "Subset Editor", you have a preview area on the right, which displays all subset-modifications at once, provided that the option "Auto" is activated. If the option "Auto" is not activated, then you can update the preview by clicking on the button shown on the right. In the lower left corner you can see the PALO.SUBSET-function and its sub-functions and how they change during the creation of a subset. With the "Paste Into"-button, the subset-function will be inserted into the Excel-table.

**Please mark a sufficiently large area at first, in order that the display of the subset will not get truncated!**

The settings in the register-cards "General" and "Sort" will always be taken over to the subset – other filters have to be activated explicitly (red arrow):
With the filters from "Hierarchy" up to "Data", check marks indicate which of these filters are activated:

Subset-settings can be saved only for current user or for all users. In order to do that, click on "Local Subsets" or on "Global Subsets" first. Then click on the button "New Subset", give the subset a name and finally click on "Save" to save the settings.
If you have saved a subset, you can paste it later elsewhere. That proceed as follows:
1. You select a sufficient large area in the spreadsheet for the subset.
2. Then you call the subset editor.
3. You select the desired database.
4. You select the dimension for which the subset is stored.
5. You select the group Local Subsets or Global Subsets.
6. Select the stored subset.
7. Paste this subset.

P.S.: The button "Formula Subset" is planned for future expansions.

For the purpose of demonstration, we always use the data base "Demo" and the dimension "products" in the following examples.

Example 1:

<table>
<thead>
<tr>
<th>Subset General Settings</th>
<th>Layout</th>
<th>Hierarchy enumeration</th>
<th>Optional Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show parents below children</td>
<td>Without effect</td>
<td>Without effect</td>
<td></td>
</tr>
</tbody>
</table>
Example 2:

<table>
<thead>
<tr>
<th>Subset General Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Layout</strong></td>
</tr>
<tr>
<td>Hierarchy</td>
</tr>
</tbody>
</table>

Example 3:

<table>
<thead>
<tr>
<th>Subset General Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Layout</strong></td>
</tr>
<tr>
<td>Show parents below children</td>
</tr>
</tbody>
</table>

More examples follow in the next sub-chapter.
12.3 Details of PALO.SUBSET()

With a number of examples, we will explain, how the PALO.SUBSET()-function and the filters belonging to it are used.

The principal structure of this function is:

\[ =\text{PALO.SUBSET}(\text{Server/Database,Dimension,Indent,Alias,Filter1,Filter2,Filter3,Filter4,Filter5,Filter6}). \]

The filters1 to 6 represent the already mentioned filter functions:

- PALO.HFILTER (Hierarchies)
- PALO.TFILTER (Texts)
- PALO.PICKLIST (manual list of elements)
- PALO.AFILTER (Attributes)
- PALO.DFILTER (Data, e.g. all values)
- PALO.SORT (Sorting)

PALO.SUBSET() is an array-function which means, that you have to copy it into a cell area after you have created it and then you must activate the array-functionality. Make sure that the cell area includes enough cells to display all elements of the subset. In order to create a subset without the help of the Subset Editor, proceed as follows:

1. Create the function with the function-assistant of Excel.
2. Copy the function to the area, in which the subset shall be displayed.
3. Mark this area.
4. In the formula bar, click behind the function.
5. Then press **CTRL+SHIFT+ENTER**, in order to activate the array-functionality. The key combination **CTRL+CAPSLOCK+ENTER** deactivates the array-functionality.
Below you can see a base-subset. If no sorting-filter is specified, elements will be displayed in the order of creation. Take a look at the structure of the function in the formula bar. There you can see the base-filter that is the pure Subset part: =PALO.SUBSET(Server/Database, Dimension, Indent, Alias). In this example, the "Alias" argument is left empty.

Example:

```excel
=PALO.SUBSET(SA$1, $C$2, $C$3)
```

This handbook has been personalized for NA - ohneanmeldung
### 12.3.1 Indent

In order that the indent-information gets displayed, the array function must extend over three columns:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>localhostDemo</td>
<td>Sales</td>
<td>Dimension</td>
</tr>
<tr>
<td>3 Europe</td>
<td>Products</td>
<td></td>
</tr>
<tr>
<td>4 Year</td>
<td>Indent</td>
<td></td>
</tr>
<tr>
<td>5 2005</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 Budget</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**IN A9:C38: [=PALO.SUBSET($A$1;$C$2;$C$5)]

| 9    | 3 Desktop L                          |
| 10   | 3 Desktop Pro                         |
| 11   | 3 Desktop Pro XL                      |
| 12   | 3 Desktop High XL                     |
| 13   | 3 Desktop High XG                     |
| 14   | 3 Server Power XC                     |
| 15   | 3 Server Power TT                     |
| 16   | 3 Server Dual C                       |
| 17   | 3 Server Dual XC                      |
| 18   | 3 Server Lion RX                      |
| 19   | 3 Notebook SX                         |
| 20   | 3 Notebook GT                         |
| 21   | 3 Notebook LXC                        |
| 22   | 3 Notebook TT                         |
| 23   | 3 Notebook SL                         |
| 24   | 3 Subnote SL                          |
| 25   | 3 Subnote XC                          |
| 26   | 3 TFT Monitor XA                       |
| 27   | 3 TFT Monitor TL                      |
| 28   | 3 TFT Monitor XP                      |
| 29   | 3 Wireless Mouse XT                   |
| 30   | 3 Optical Mouse TS                    |
| 31   | 3 Laptop Mouse S                      |
| 32   | 3 Keyboard XA                         |
| 33   | 3 Keyboard GT                         |
| 34   | 2 Stationary PC’s                     |
| 35   | 2 Portable PC’s                       |
| 36   | 2 Monitors                            |
| 37   | 2 Peripherals                         |
| 38   | 1 All Products                        |
Depending on the argument 1, 2 or 3, a different hierarchy-numbering will be used. It affects the values in the indent column. For the filter-displays themselves, the indent argument only takes effect, if the hierarchy-level has been specified in a subset-filter.

**Indent 1:**
With indent 1 (or empty), the hierarchy-numbering of the elements is: 3, 2, 1

In general: The highest hierarchy-level gets the number 1, the second highest 2 and so on.

**Indent 2:**
With indent 2, the hierarchy-numbering of the elements is: 0, 1, 2

In general: Elements in the lowest hierarchy-level (base elements) get the number 0, the number is incremented by 1, for every step up in the hierarchy.

**Indent 3:**
With indent 3, the hierarchy-numbering of the elements is: 2, 1, 0

In general: The highest hierarchy-level get the number 0, the second highest 1 and so on.
12.3.2 Alias

So far, you have created a subset without utilizing the argument "Alias". Now we would like to supplement the subset with Alias and compare the effect. To this end, we create the following subset in cell C9:

=PALO.SUBSET($A$1,$C$2,$C$5)

Copy this function into the area C9:C25 and then activate the array-functionality (Ctrl+Shift+Enter). Enter "Alias" into cell E1 and "German" into cell E2. Copy the area C9:C25 to the area E9:E25. Deactivate the array-functionality for the area E9:E25 (Ctrl+Capslock+Enter), then mark E9. In the formula bar, click inside the PALO.SUBSET()-function. Now click on the function-assistant. It opens and you can now supplement the entry for "Alias" with $E$2.

When using "Alias", the following applies: The Alias impacts the Text Filter and the Sort Filter indirectly. It will be applied to the Text Filter automatically. In regards to the Sort Filter, a sort by Aliases can be activated separately.

In other words: Aliases will not be displayed directly, but will only be used as sort criteria. In case you would like to see the Aliases, you must extend the output area of the matrix formula to 2 columns.

Two attributes, which function as Aliases, may be specified at a maximum. In that case, the second attribute will be searched first. In case the corresponding cell in the attribute cube is empty, the first attribute will be searched. Should that cell be empty as well, the original name of the element will be used instead.

If you want to make the aliases visible, you must extend the return-area of the array-function to two columns.

Therefore, copy this function into the area E9:F25. Leave the area marked and click in the formula bar. Then activate the array-functionality for this area by pressing Ctrl+Shift+Enter.

Confirm the changes with "OK".
Let's update our demo-application at first: We already have created the attributes "German", "English" and "French" for the dimension "Months". As well as the attribute "Color" for the dimension "Products" (see chapter: "Working with Attributes"). Create the following view and supplement missing entries for our further examples:

At first, we will sort using the attributes of the dimension "Months".

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>localhost/Demo</td>
<td>localhost/Demo</td>
<td>localhost/Demo</td>
<td>localhost/Demo</td>
</tr>
<tr>
<td># Months</td>
<td># Months</td>
<td># Months</td>
<td># Months</td>
</tr>
<tr>
<td>Year</td>
<td>Jahr</td>
<td>Year</td>
<td>année</td>
</tr>
<tr>
<td>Qtr. 1</td>
<td>Quartal 1</td>
<td>quarter 1</td>
<td>trimeste 1</td>
</tr>
<tr>
<td>Jan</td>
<td>Januar</td>
<td>January</td>
<td>janvier</td>
</tr>
<tr>
<td>Mar</td>
<td>März</td>
<td>March</td>
<td>mars</td>
</tr>
<tr>
<td>Qtr. 2</td>
<td>Quartal 2</td>
<td>quarter 2</td>
<td>trimeste 2</td>
</tr>
<tr>
<td>Apr</td>
<td>April</td>
<td>April</td>
<td>avril</td>
</tr>
<tr>
<td>May</td>
<td>Mai</td>
<td>May</td>
<td>mai</td>
</tr>
<tr>
<td>Jun</td>
<td>Juni</td>
<td>June</td>
<td>juin</td>
</tr>
<tr>
<td>Qtr. 3</td>
<td>Quartal 3</td>
<td>quarter 3</td>
<td>trimeste 3</td>
</tr>
<tr>
<td>Jul</td>
<td>Juli</td>
<td>July</td>
<td>juillet</td>
</tr>
<tr>
<td>Aug</td>
<td>August</td>
<td>August</td>
<td>août</td>
</tr>
<tr>
<td>Sep</td>
<td>September</td>
<td>September</td>
<td>septembre</td>
</tr>
<tr>
<td>Qtr. 4</td>
<td>Quartal 4</td>
<td>quarter 4</td>
<td>trimeste 4</td>
</tr>
<tr>
<td>Oct</td>
<td>Oktober</td>
<td>October</td>
<td>octobre</td>
</tr>
<tr>
<td>Nov</td>
<td>November</td>
<td>November</td>
<td>novembre</td>
</tr>
<tr>
<td>Dec</td>
<td>Dezember</td>
<td>December</td>
<td>décembre</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>localhost/Demo</td>
<td>localhost/Demo</td>
</tr>
<tr>
<td># Products</td>
<td># Products</td>
</tr>
<tr>
<td>Color</td>
<td>Color</td>
</tr>
<tr>
<td>Desktop L</td>
<td>black</td>
</tr>
<tr>
<td>Desktop Pro</td>
<td>silver</td>
</tr>
<tr>
<td>Desktop Pro XL</td>
<td>blue</td>
</tr>
<tr>
<td>Desktop High XL</td>
<td>black</td>
</tr>
<tr>
<td>Desktop High XQ</td>
<td>silver</td>
</tr>
<tr>
<td>Server Power XC</td>
<td>blue</td>
</tr>
<tr>
<td>Server Power TT</td>
<td>black</td>
</tr>
<tr>
<td>Server Dual C</td>
<td>silver</td>
</tr>
<tr>
<td>Server Dual XC</td>
<td>blue</td>
</tr>
<tr>
<td>Server Lion RX</td>
<td>black</td>
</tr>
<tr>
<td>Notebook SX</td>
<td>silver</td>
</tr>
<tr>
<td>Notebook GT</td>
<td>blue</td>
</tr>
<tr>
<td>Notebook LXC</td>
<td>black</td>
</tr>
<tr>
<td>Notebook TT</td>
<td>silver</td>
</tr>
<tr>
<td>Notebook SL</td>
<td>blue</td>
</tr>
<tr>
<td>Subnote SL</td>
<td>black</td>
</tr>
<tr>
<td>Subnote XK</td>
<td>silver</td>
</tr>
<tr>
<td>TFT Monitor YA</td>
<td>blue</td>
</tr>
<tr>
<td>TFT Monitor TL</td>
<td>black</td>
</tr>
<tr>
<td>TFT Monitor XP</td>
<td>silver</td>
</tr>
<tr>
<td>Wireless Mouse XT</td>
<td>blue</td>
</tr>
<tr>
<td>Optical Mouse TS</td>
<td>black</td>
</tr>
<tr>
<td>Laptop Mouse S</td>
<td>silver</td>
</tr>
<tr>
<td>Keyboard XX</td>
<td>blue</td>
</tr>
<tr>
<td>Keyboard GT</td>
<td>black</td>
</tr>
</tbody>
</table>

At first, we will sort using the attributes of the dimension "Months".
The array-function for the table displayed above is:

\[
\begin{align*}
C9:C38: \{=\text{PALO}.\text{SUBSET}(\$A\$1, \$C\$2, \$C\$5)\} \\
E9:F25: \{=\text{PALO}.\text{SUBSET}(\$A\$1, \$C\$2, \$C\$5, \$E\$2)\} \\
H9:J38: \{=\text{PALO}.\text{SUBSET}(\$A\$1, \$C\$2, \$C\$5, \$H\$5, \text{PALO}.\text{SORT}(3), \text{PALO}.\text{TFILTER}({"a\*","j\*","s\*","t\*"}))\} \\
K9:L38: \{=\text{PALO}.\text{SUBSET}(\$A\$1, \$C\$2, \$C\$5, \$K\$5, \text{PALO}.\text{SORT}(3))\} \\
N9:O25 \\
\{=\text{PALO}.\text{SUBSET}(\$A\$1, \$C\$2, \$C\$5, \$N\$2, \text{PALO}.\text{SORT}(3), \text{PALO}.\text{TFILTER}({"a\*","j\*","s\*","t\*"}))\}
\end{align*}
\]

In order to get the aliases displayed, the array-function has to be entered in two columns. The comparison of two different aliases shows the changes in the sorting. Because the sorting-argument is "3=ALIAS" in each case, column H is sorted using German names, column K is sorted using English names and column N is sorted using French names.

**Sorting using ASCII-Code:** In column L you can see, that the sorting is done using ASCII-code, i.e. **upper case letters precede lower case letters**!

In the current example you can also see, how the usage of "Alias" affects the TFilter. As you can see, the elements "Qtr.*" and "year", for example, are kept, because their aliases are respectively "trimestre" and "année". These elements are correctly recognized by the Text-Filter.
12.3.3 Filter Functions

You have learned how the basis function of the subset works. Now you will learn which arguments exist for the various filters and how to apply them.

12.3.3.1 PALO.HFILTER()

PALO.HFILTER() defines hierarchically a subset with criteria relative to 1. ELEMENT or 2. REVOLVE_ELEMENT. Case 1 and Case 2 should not be combined.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Element</td>
<td>Element name</td>
<td>Activates the selection of elements which are above or below of &quot;Element&quot; within the dimension hierarchy</td>
</tr>
<tr>
<td>Above</td>
<td>TRUE/FALSE</td>
<td>This argument can only be used in combination with &quot;Element&quot;. If TRUE, the elements will be selected that are above the &quot;Element&quot; within the hierarchy. Otherwise the elements below (FALSE or BLANK). Above / Below means that the elements have to be direct or indirect children or parents of &quot;Element&quot; respectively.</td>
</tr>
<tr>
<td>Exclusive</td>
<td>TRUE/FALSE</td>
<td>This argument can only be used in combination with &quot;Element&quot;. TRUE = Do not show the element itself, i.e. Element indicated in &quot;Element&quot;. Blank or FALSE = Show the element.</td>
</tr>
<tr>
<td>Hide</td>
<td>Number</td>
<td>This argument can only be used in combination with &quot;Element&quot;. It hides either the base elements (HIDE = 1) or the consolidations (HIDE = 2). Blank = nothing will be hidden.</td>
</tr>
<tr>
<td>Revolve element</td>
<td>Element name</td>
<td>The effect of this argument is that it will remove elements from the subset, if elements are not on the same level as &quot;Revolve element&quot;. Therefore this argument should not be used in combination with &quot;Element&quot;. Blank = None. &quot;Revolve element&quot; additionally requires the specification of &quot;Revolve count&quot;. Otherwise, an error message will be returned.</td>
</tr>
<tr>
<td>Revolve count</td>
<td>Number</td>
<td>Determines how long the revolving list should be. The length of the list is dictated by the number in &quot;Revolve count&quot;. Blank = None.</td>
</tr>
<tr>
<td>Revolve add</td>
<td>Number</td>
<td>At first, display elements at a lower level (= 1) or at a higher level (=2), respectively. Here, the level-fields of the elements, assigned by the Palo Server, are used as criterion. &quot;Revolve add&quot; additionally requires the specification of &quot;Revolve count&quot;. Otherwise, an error message will be returned. Blank = nothing.</td>
</tr>
<tr>
<td>Level start</td>
<td>Number</td>
<td>Elements that do not meet the level entered in &quot;Level start&quot; at a minimum and the level in &quot;End level&quot; at a maximum, will be removed.</td>
</tr>
<tr>
<td>End level</td>
<td>Number</td>
<td>2nd argument belonging to &quot;Level start&quot;.</td>
</tr>
</tbody>
</table>
The following example illustrates the use of "Element". Let's first create the subset without utilizing the filters.

Then add the filters by entering the information for the HFilter and the argument "Above" in cells "E1:E2" and "E4:E5". You will be able to track the changes, if you demonstrate the utilization of the HFilter in another column. Therefore, please copy the range "A8:A38" to "C8:C38". Now click on cell "C9" and then in the formula bar.

You can now call the function-assistant with \( \text{fx} \) and then enter the arguments for \( \text{PALO.HFILTER()} \).

**Please note:** It is **not** possible to create this sub function as an independent function e.g. in cell "G2" and then reference the same in \( \text{PALO.SUBSET()} \).

Entering

\[ =\text{PALO.SUBSET}(\text{A1}, \text{C2}, \text{C5}, \text{G2}) \]

will cause an error((#value!)).

Therefore you have to enter the filters directly in the \( \text{PALO.SUBSET} \)-function. However, you are required to exactly know the syntax of the particular filter. Only then it is possible that you correctly fill e.g. "HFilter()".

The simpler way is to first enter the name of the function with empty braces:
After you have done that, please close the Function-Assistant by clicking OK. Now you can select the cell E9 anew and then click on PALO.HFILTER() in the formula bar, in order to restart the Function-Assistant again with \( fx \):

With the Function-Assistant you now get the function PALO.HFILTER() and you can enter the arguments for this function, e.g. E2.

1. Filter elements of a subset by defining struct criteria. 2. Repeating a subset with a different structure. Case 1 and Case 2 cannot be combine.
For the next subset, we would like to incorporate the argument "ABOVE". Open the Function Assistant again, enter "TRUE" and close the Assistant.

Please proceed accordingly with the arguments "Exclusive" and "Hide". Exclusive = TRUE will not show "Monitors". Hide = 1 hides all base elements whereas Hide = 2 hides the consolidated elements.
Below overview illustrates how the output of the elements is changing by utilizing "HFilter". By entering "Monitors", all elements are omitted which are not below Monitors from a hierarchy point of view. This is reverted by using Above = "TRUE". Only elements that are >= Monitors will be displayed.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>localhost/Demo</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Sales</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Europe</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Year</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2005</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Budget</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Exclusive = "TRUE" shows respectively the elements of PALO.HFILTER($E$2) or PALO.HFILTER("Monitors"). However, "TRUE" removes the entry "Monitors", whereas "Hide", with the value "1", hides the corresponding base elements.

**NOTE:** "Above", "Exclusive" and "Hide" only work if "Element" is filled in.

In the previous examples we illustrated the selections based on hierarchy, which is steered via the argument "Element" along with other associated arguments. The following section covers hierarchical selections which do not permit the use of the argument "Element".

The first example illustrates the use of **Revolve_Element** and its sub-arguments:

- **REVOLVE_ELEMENT** ("Monitors") determines that elements, which are not on the same level as "Monitors", are not displayed.
- **REVOLVE_COUNT** determines the length of the list to be shown and with that the element repetitions if applicable.
- **REVOLVE_ADD** determines if elements with a higher (1) or lower (2) level are to be shown as well.
To prepare the example, go back to the standard subset once again. Please add the possible arguments in the header section up until column "K". Afterwards, the simple subset should be copied accordingly. The HFilter will be adjusted as described below:

- \( C9:C38: \{=\text{PALO.SUBSET}(\$A1,\$C2,\$C5,\text{PALO.HFILTER}(E2)) \}\)
- \( E9:E38: \{=\text{PALO.SUBSET}(\$A1,\$C2,\$C5,\text{PALO.HFILTER}(E2,\ldots,\text{E5},\text{G2},\text{G5})) \}\)
- \( G9:G38: \{=\text{PALO.SUBSET}(\$A1,\$C2,\$C5,\text{PALO.HFILTER}(\ldots,\text{E5},\text{G2},\text{G5})) \}\)
- \( I9:I38: \{=\text{PALO.SUBSET}(\$A1,\$C2,\$C5,\text{PALO.HFILTER}(\ldots,\text{E5},\text{I2},\text{I5})) \}\)
- \( K9:K38: \{=\text{PALO.SUBSET}(\$A1,\$C2,\$C5,\text{PALO.HFILTER}(\ldots,\text{E5},\text{K2},\text{K5})) \}\)

This is the result:

```
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>&quot;Subset Basic&quot;</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>K1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Desktop L</td>
<td>TFT Monitor YA</td>
<td>Monitors</td>
<td>Stationary PCs</td>
<td>Desktop L</td>
<td>Stationary PCs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Desktop Pro</td>
<td>TFT Monitor TL</td>
<td>Monitors</td>
<td>Portable PCs</td>
<td>Desktop Pro</td>
<td>Portable PCs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Desktop Pro XL</td>
<td>TFT Monitor XP</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Desktop Pro XL</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Desktop High XL</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Desktop High XL</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Desktop High XG</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Desktop High XG</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Server Power XC</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Server Power XC</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Server Power TT</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Server Power TT</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Server Dual C</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Server Dual C</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Server Dual XC</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Server Dual XC</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Server Lion RX</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Server Lion RX</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Notebook SX</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Notebook SX</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Notebook GT</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Notebook GT</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Notebook LVC</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Notebook LVC</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Notebook TT</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Notebook TT</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Notebook SL</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Notebook SL</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Sunnite SL</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Sunnite SL</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Subnote XK</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Subnote XK</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>TFT Monitor YA</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>TFT Monitor YA</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>TFT Monitor TL</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>TFT Monitor TL</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>TFT Monitor XP</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>TFT Monitor XP</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Wireless Mouse XT</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Wireless Mouse XT</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Optical Mouse TS</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Optical Mouse TS</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>Laptop Mouse S</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Laptop Mouse S</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>Keyboard XX</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Keyboard XX</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>Keyboard GT</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Keyboard GT</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>Stationary PCs</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Stationary PCs</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>Portable PCs</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Portable PCs</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>Peripherals</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Peripherals</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>All Products</td>
<td>Monitors</td>
<td>Monitors</td>
<td>Monitors</td>
<td>All Products</td>
<td>Monitors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

The Subset Basic is shown in column "A". In column "C" you see once again the simple HFilter with "Monitors" in "Elements". We mentioned that the arguments "Elements" and "Revolve_Element" may not be used in combination. Column "E" illustrates the result if you do it nevertheless:

The argument "Element" (Monitors = E2) restricts the selection of the elements to "Monitors" and the subjacent elements. The argument "Revolve_Element" determines that elements, which are not on the same level as "Monitors", shall not be displayed. However, "Revolve_Element" would show the elements that are on the same level (e.g. Peripherals). The combination of "show me everything that is on the same level as "Monitors"" and "show me "Monitor" and everything below that results in a conflict. In the
end, leaves "Monitor", which – due to the argument "Revolve_Count" - is being repeated until 15 elements are displayed.

The next examples in columns "G" to "K" illustrate the correct use of the argument "Revolve_Element" in various combinations with the associated arguments "Revolve_Count" and "Revolve_Add".

In G9:G38:
Palo.hfilter(,,,$E$5,$G$2,$G$5) omits all elements, that are not on the same level as "Monitors. Since it reads "Revolve_Add" = 1, the levels above that will be included. Consequently, "All Products", a higher level element, will be displayed. Due to "Revolve_Count" = 15, the elements are being repeated until 15 elements are displayed.

In I9:I38:
Palo.hfilter(,,E5,I2,I5) effects that, because of the change "Revolve_Add" = 1, "Monitors" and all elements below the level of Monitors will be displayed. Consequently, all elements at the same or at a lower level than "Monitors" are displayed. Due to "Revolve_Count" = 30, the elements are repeated, until 30 elements are displayed. The 30th element "Desktop L" is a repetition, in order to satisfy the requirement "Revolve_Count = 30". The subset will be truncated in this case, if the value of "Revolve_Count" is less than 29.

In K9:K38:
Palo.hfilter(,,E5,I2,I5) effects that, because of the change "Revolve_Add" = 2 "Monitors" and all elements above the level of Monitors will now be displayed. Due to "Revolve_Count" = 15, the elements are repeated, until the list contains 15 elements.

Example K9:K38 with the Subset Editor:

<table>
<thead>
<tr>
<th>Database</th>
<th>Dimension</th>
<th>Layout</th>
<th>Hierarchy enumeration</th>
<th>Optional Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demo</td>
<td>Products</td>
<td>Flat</td>
<td>Without effect</td>
<td>Without effect</td>
</tr>
</tbody>
</table>

This handbook has been personalized for NA - ohneanmeldung
LEVEL_START" and "END_LEVEL" should also not be combined with the argument "Element". Neither should they be combined with the argument "Revolve_Element". Once again, let's take the simple subset as a starting point and add the arguments for "Level_Start" / "End_Level". The combination of the two "level"-indications restricts the output accordingly.

The array-functions are:

F9:F38: \{=PALO.SUBSET($A$1,$C$2,$C$5,,PALO.HFILTER(,,,,,,E2,E5))\}

H9:H38: \{=PALO.SUBSET($A$1,$C$2,$C$5,,PALO.HFILTER(,,,,,,G2,G5))\}

The PALO.HFILTER(,,,,,,1,2) starts at the level of "Monitors" and ends at the level "All Products" (column "E").

Correspondingly, the PALO.HFILTER(,,,,,,0,1) starts with the first base element and ends at the level of "Monitors" (column "G").

Important: The filter-display depends on the argument "Indent" here.
12.3.3.2 PALO.TFILTER()

PALO.TFILTER() remove elements, which do not match a particular string (restricting). Upper case letters and lower case letters are distinguished throughout.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regex</td>
<td>Text</td>
<td>In case of a list of regular expressions, those elements will be omitted from the subset which do not correspond with either expression in the list. Only the characters &quot;*&quot; and &quot;?&quot; are wildcards.</td>
</tr>
<tr>
<td>Extended</td>
<td>TRUE/FALSE</td>
<td>If EXTENDED = TRUE, Perl regular expressions will be used. Signs such as &quot;[&quot; or &quot;]&quot; have an impact. If EXTENDED = blank, the regular expression will be handled as simple DOS expressions and only &quot;*&quot; and &quot;?&quot; have an impact.</td>
</tr>
</tbody>
</table>

Examples of regular expressions in PERL:
- ^K = Everything that begins with a "K".
- X$ = The expression ends with an X.
- X* = X followed by an arbitrary number of characters.
- XP+ = "XP" must occur at least once but may occur several times.
  (Upper case and lower case letters are distinguished!)
E9:E38:
In this TFilter, we use simple expressions which are well known from the DOS console.
PALO.TFILTER(\"Server\", \"N\", \"O\") resp. PALO.TFILTER($E$2:$E$5) remove all elements, which don’t match the pattern \{"Server\", \"N\", \"O\"\}.

G9:G38:
Here we use the Perl-regular expressions. PALO.TFILTER(\"X\", [s,d], True) or PALO.TFILTER($G$2:$H$2,True) keep all elements which end on an "X" or contain an "s" or a "d".

This handbook has been personalized for NA - ohneanmeldung
Example E9:E38 with the Subset Editor:

<table>
<thead>
<tr>
<th>Database</th>
<th>Dimension</th>
<th>Layout</th>
<th>Hierarchy enumeration</th>
<th>Optional Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demo</td>
<td>Products</td>
<td>Flat</td>
<td>Without effect</td>
<td>Without effect</td>
</tr>
</tbody>
</table>

**12.3.3.3 PALO.PICKLIST()**

With PALO.PICKLIST(), you can either add elements to the subset, which then by no means will be removed. Or you can determine a set of elements as pre-selection – only these elements will be passed on to further filters of the subset.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Definition</td>
<td>Name of elements</td>
<td>Which elements belong to the pick list. The definition is essential to the pick list.</td>
</tr>
<tr>
<td>Type</td>
<td>Number</td>
<td>0/empty = Insert before the list, 1 = Insert after the list, 2 = insert into the list, 3 = As pre-selection for the subset.</td>
</tr>
</tbody>
</table>
Starting with our example PALO.TFILTER(), modify and complete the entries in the rows 1-8:

G9:G38:

\[
\{=PALO.SUBSET($A$1,$C$2,$C$5,,PALO.TFILTER($E$2:$E$4),PALO.PICKLIST($G$2:$G$4,$G$6))\}
\]

In columns I to M, only the pick list-presentation changes, i.e. the argument $G$6!
Example I9:I38 with the Subset Editor:

<table>
<thead>
<tr>
<th>Database</th>
<th>Dimension</th>
<th>Layout</th>
<th>Hierarchy enumeration</th>
<th>Optional Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demo</td>
<td>Products</td>
<td>Flat</td>
<td>Without effect</td>
<td>Without effect</td>
</tr>
</tbody>
</table>

With the Subset Pick list-Filter, you can select the desired elements by pushing the "Edit"-button:
12.3.3.4 PALO.AFILTER()

PALO.AFILTER() removes elements that do not match defined attributes.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Advanced Filter</td>
<td>Text-Array</td>
<td>In case a table containing attributes is rendered, only those elements will</td>
</tr>
<tr>
<td></td>
<td></td>
<td>be selected, the attributes of which correspond with either row in the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>matrix. The table's first column contains the name of the attribute. The</td>
</tr>
<tr>
<td></td>
<td></td>
<td>criteria next to it. The following operators for numeric criteria are</td>
</tr>
<tr>
<td></td>
<td></td>
<td>supported: &lt;, &gt; and '!='. Please pay attention to '*' in front of '='!</td>
</tr>
</tbody>
</table>

The element which you wish to select, must correspond to one of the rows in the selection-array G1:H4). If this is the case, the element will be displayed. Otherwise it will be hidden. Therefore, all the elements with the colour "blue", which cost more than "1500", all the elements with the colour "silver", which cost less than "500", and all the elements with the colour "black", which cost exactly "2000", will be displayed. All other elements will not be displayed. The result can be seen in column "G":

G9:G29

{=PALO.SUBSET($A$1,$E$2,$E$5,,PALO.AFILTER($G$1:$H$4))}

This handbook has been personalized for NA - ohneanmeldung
### 12.3.3.5 PALO.DFILTER()

PALO.DFILTER() calculates a value for each of the elements of the subset. It is either a numeric value or a string value. In case of numeric values, a slice can be defined for each of the elements of the subset. The slice is always the same except for the element itself. Now you can do operations using the cell values within the slices, such as Total Sum, Maximum or Minimum based on all cells. This results in a value for each of these elements.

The elements can now be filtered based on their values, whereby operations such as "<100" will be applied.

The filtering is similar for string values, e.g. you could filter by applying ">w", whereas lexicographical rules apply. However, the slice may only contain one cell. The reason is obvious: You cannot summarize strings.

---

**Example G9:G29 with the Subset Editor:**

<table>
<thead>
<tr>
<th>Subset General settings</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Database</strong></td>
</tr>
<tr>
<td>Demo</td>
</tr>
</tbody>
</table>

**Subset Attribute Filter**

```
<table>
<thead>
<tr>
<th>Price per Unit</th>
<th>Products</th>
<th>Color</th>
<th>English</th>
<th>German</th>
<th>Other Attributes</th>
<th>Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt;1500</td>
<td>blue</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&lt;500</td>
<td>silver</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>&gt;2000</td>
<td>black</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Preview**

- Server DualXPC
- Notebook 6T
- Subnote SL
- TFT Monitor XP
- Laptop Mouse S
<table>
<thead>
<tr>
<th><strong>Argument</strong></th>
<th><strong>Form</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>PALO.SUBCUBE()</td>
<td>Function</td>
<td>Palo.Subcube is an essential selection criterion when using DFilter. The arguments are the name of the cube and the dimensions excluding the dimension which is covered by the subset. The elements are entered in the sequence, in which the dimensions are defined in the cube. The place in the dimension, which is already determined by the subset, remains empty.</td>
</tr>
</tbody>
</table>
| Operations | Take a look at the examples: | It is checked, whether the values determined for the elements satisfy the criteria. Possible operators: \(=,<,>,\leq,\geq,<>\)  
Example 1 for a data value: \(\geq\).3000\}  
Example 2 for a string: \(=\)."Super") (Here, "Super" is an entry in a text-element) |
| Top | Number | This is another possible argument. In case an integer value TOP is passed, only a number of TOP elements with the highest values are displayed. If the argument is left blank, the elements will not be restricted. |
| Upper percentage | Number | If a numeric value between 1 and 99 (inclusive) is entered here, only those elements will be selected, if their cumulated value results in a total value that is below the marginal value. E.g. 50%: All cumulated values are being evaluated that are \(\leq\) 50%. Once the value exceeds 50%, the selection ends. The selection begins with the highest element. |
| Lower percentage | Number | If a numeric value between 1 and 99 (inclusive) is entered here, only those elements will be selected, if their cumulated value results in a total value that is below the marginal value. E.g. 5%: Cumulated values up until 5% will be evaluated. Once the value exceeds 5%, the selection ends. The selection begins with the lowest element.  
If both Percentage-arguments are used, the "Mid-percentage" applies. I.e. the values up until the highest or the lowest marginal value respectively will be removed from the selection. Taking above example, all values between \(>50%\) (starting with the highest value) and \(>5%\) (starting with the lowest value) will be included in the selection. |
| Cell Operator | Number | If a value is entered, the respective operator will be applied to each slice. You can e.g. calculate the sum, minimum or maximum of the cells in the slice. 6 = STRING expects strings to be contained in the cells. For slice use 1. Operators: 0/blank = SUM, 1=ALL, 2=AVERAGE, 3=MAXIMUM, 4=ANY, 5=MINKIMUM, 6=TEXT  
Returned are the elements – not the values. |
| No Rules | True/False | Empty/false = Enterprise-rules are computed for DFilters. True = Enterprise-rules are not computed for DFilters. |
A simple example illustrates how to display from a list of products only the Top 10 in respect to the number of Units. To demonstrate this, let's start with our standard subset. Add the information shown in range "E1:F5". In addition to the original data, please add the element "Units" in "C8". Please complete "C9:C38" by using the "Paste Data Function", "Palo.Data" and "Guess Arguments" from the "Sales" cube.

You will get the following view:

![Microsoft Excel - Subset_DFilter.xls](image-url)
Now please copy the ranges "A8:A38" and "C8:C38" to "E8:F38". Change the text in "E8" to "Products (DFilter)". In "F8" please change the parameter from "$A9" to "$E9". Afterwards please copy the adjusted formula to "F10:F38". In order to modify E9, we turn off the array-functionality in the area E9:E38 (Ctrl+Capslock+Enter). To achieve this, click on E9 and then on \( \text{fx} \):

The Assistant will open the window that allows you to enter the parameters.

In Filter1 please enter the function PALO.DFILTER() and confirm by pressing "OK".

Now click onto PALO.DFILTER() in the formula bar and repeat the process. Now you can enter the arguments for PALO.DFILTER() in the Function-Assistant.
Finally, you have to define the "Subcube". Please proceed in the same manner as you did when adding \texttt{PALO.DFILTER()}.

\texttt{PALO.SUBCUBE()} consists of the cube "Sales" and the dimensions of the cube. The dimension that affects the subset ("Products") remains blank.

Therefore, the subset reads: \texttt{PALO.SUBCUBE("Sales", "Europe", "Year", "2007", "Actual", "Units")} resp. \texttt{PALO.SUBCUBE($A$2, $A$3, $A$4, $A$5, $A$6, $F$8)}.
In the end, you will get a view of the Top 10, i.e. those elements are selected that meet the requirement "Units > 300000". The elements are displayed in the order of creation.

E9:E38:

\(=\text{PALO.SUBSET}(\text{A1},\text{C5},\text{C5},,\text{PALO.DFILTER(\linebreak PALO.SUBCUBE(\text{A2},\text{A3},\text{A4},\text{A5},\text{A6},\text{F8}),\text{E2:F2}),\text{E5})),\text{PALO.SORT(1,1)})\)

Finally, we copy E9:E38 to H9:H38 and change the sorting:

H9:H38:

\(=\text{PALO.SUBSET}(\text{A1},\text{C5},\text{C5},,\text{PALO.DFILTER(\linebreak PALO.SUBCUBE(\text{A2},\text{A3},\text{A4},\text{A5},\text{A6},\text{S8}),\text{E2:F2}),\text{E5}),\text{PALO.SORT(1,1)})\)}
Example H9:H38 with the Subset Editor:

<table>
<thead>
<tr>
<th>Database</th>
<th>Dimension</th>
<th>Layout</th>
<th>Hierarchy enumeration</th>
<th>Optional Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demo</td>
<td>Products</td>
<td>Hierarchy</td>
<td>Without effect</td>
<td>Without effect</td>
</tr>
</tbody>
</table>

Subset General settings

Subset Sort: By Value

The computations done by the PALO.DFILTER() are very time consuming. Therefore, the auto-update feature of the preview is turned off by default, if the data-filter is activated.

Also keep in mind, that all dimensions must be specified. Using wildcards ("*"*) in PALO.SUBLCUBE() results in errors.

```
=PALO.SUBSET("localhost/Demo","Products","1",PALO.DFILTER(PALO.SUBLCUBE("Sales","*","*","*","*","*","*",0),PALO.SORT(0,0,0,0,1)))
```
As further example, we now produce a PALO.DFILTER with "Lower_percentage".

By the declaration of this argument, only the elements with the lowest value are chosen. Which means, that their cumulated values correspond to 2% of the total sum over all values.

To this end, we change the preceding example as follows: Instead of using "Top", we now use a "Lower_percentage" of 2% in the cells "E4:E5". This has the effect, that the elements of lowest value are displayed, until their sum exceeds 2% of the total sum of all elements.

The following picture shows the connection. In cell K17, the value of 2% is exceeded.

The element, with which the limit is exceeded, is added to the selection.
Take the previous example with PALO.DFILTER() and store it under a new name, then delete the area E1:F5 and create the entries as in the following screen-shot. Cancel the array-functionality for the area E9:E38 (Ctrl+Capslock+Enter) and change the entry in E9 as follows:

\[=\text{PALO.SUBSET(}A1,A2,A3,A4,A5,F8,\text{PALO.DFILTER(}\text{PALO.SUBCUBE(}A2,A3,A4,A5,A6,F8),,,E5)\text{)}\]

This alteration of E9, based on the previous subset-function, is achieved in the simplest way, by clicking on the DFILTER in the formula bar and then starting \( \text{fx} \):
Then copy the function from E9 to E38 and reactivate the array-functionality (Ctrl+Shift+Enter).

Example E9:E38 with the Subset Editor:

<table>
<thead>
<tr>
<th>Database</th>
<th>Dimension</th>
<th>Layout</th>
<th>Hierarchy enumeration</th>
<th>Optional Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demo</td>
<td>Products</td>
<td>Flat</td>
<td>Without effect</td>
<td>Without effect</td>
</tr>
</tbody>
</table>

This handbook has been personalized for NA - ohneanmeldung
12.3.3.6 PALO.SORT()

PALO.SORT() is a structuring filter. It can, for example, sort alphabetically or use the data-values for sorting. In addition, it can produce a hierarchical order, which takes parallel hierarchies in the subset into account. Without the sorting filter, the elements are displayed in the order of creation.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Form</th>
<th>Description</th>
</tr>
</thead>
</table>
| Whole            | Number  | 0/empty = order of creation.  
|                  |         | 1 = Builds a hierarchic order and shows the children of elements, the elements which have been removed from the subset. It is a prerequisite that the children are contained in the subset. 
|                  |         | 2 = Will not show the children, but will cut the view at those particular positions. |
| Criteria         | Number  | Sort by 0/blank=Definition, 1=data, 2=Text, 3=Alias |
| Attribute        | Text    | Does not sort by Aliases if defined, but by the terms defined by the attribute. Note: This filter should not be used in combination with "Criteria"! |
| Type_Limitation  | Number  | 0/blank = no impact, 1 = sorts base elements only, 
|                  |         | 2 = sorts consolidations only 
|                  |         | In order to do a sort, it is necessary to indicate the sort criteria. Therefore, this argument works only in connection with the arguments "Whole" and "Criteria". |
| Level_Element    | Text    | If not blank, sorting takes place only at the level of this element. In order to do a sort, it is necessary to indicate the sort criteria. Therefore, this argument works only in connection with the arguments "Whole" and "Criteria". |
| Reverse          | Number  | 0/empty = without effect 
|                  |         | 1 = Hierarchical presentation of parents below the children and on every level, sorted using criteria. 
|                  |         | 2 = complete reversal of the sorting criteria. 
|                  |         | 3 = Hierarchical presentation and reversal of sorting using criteria, on every level. 
|                  |         | 1 or 3 as argument only works in conjunction with "Whole <> 0/empty". With 2 as an argument, it is allowed that "Whole = 0/empty". |
| show_duplicates  | Number  | 0/empty = don’t display duplicates 
|                  |         | 1 = display duplicates |
For the following views, please start off with the Data Filter. It is to be modified as follows:

1. Highlight PALO.SUBSET(),
2. Open the Function Assistant,
3. enter PALO.SORT() as Filter2

The display in column "E" is based on:

Filter: PALO.SORT(1,1)

With this, a hierarchy is built and the sorting is done within this hierarchy, i.e. it is sorted on each level by value ascendant, starting with the highest level "All Products". Then "Stationary PCs", "Portable PCs", "Monitors" and "Peripherals" are sorted by value ascendant. Their child elements are displayed below their parent, sorted by value ascendant.

E9:E38

{=PALO.SUBSET($A$1,$C$2,$C$5,,PALO.DFILTER(PALO.SUBCUBE($A$2,,$A$3,$A$4,$A$5,$A$6,$F$8),($E$2: $F$2),$E$5),PALO.SORT(1,1))}

In the 2nd example in column "H", the sort filter is adjusted. Please remove "1" which stands for the sorting by hierarchy. Now the sort is handled by data values.

This handbook has been personalized for NA - ohneanmeldung
In the following examples, we will treat the arguments "Attribute" and "Reverse" of PALO.SORT().

The elements of "Months" are a good example to explain the argument Attribute. This example is based on the simple subset which is supplemented with the argument "Alias". This simple subset can also be complemented by various sorting possibilities. To make a comparison, the elements of "Months" and the various attribute notations are added starting at column "G". Please pay attention to the changes in the sort pattern if the arguments "Criteria" and "Attribute" are applied.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sales</td>
<td>Dimension</td>
<td>Criteria</td>
<td>Criteria</td>
<td>Attribute</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>localhost/Demo</td>
<td>Months</td>
<td>2</td>
<td>3</td>
<td>German</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Europe</td>
<td>Year</td>
<td>Indent</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2005</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Budget</td>
<td>Alias</td>
<td>French</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This handbook has been personalized for NA - ohneanmeldung
Column D:
The simple subset is sorted by the Alias. The first element that is returned is "Year" since the corresponding French Alias is "année".

D9:D25: {=PALO.SUBSET($A$1,$B$2,$B$5,$B$7,PALO.SORT(,$D$2))}

Column E:
The simple subset is sorted by the argument "Attribute", "German" is the attribute in our case. Therefore, the sorting is done by the notations, as shown in column "I". This causes for "Year" to be inserted in front of "January" ("Year" in German is "Jahr").

E9:E25: {=PALO.SUBSET($A$1,$B$2,$B$5,$B$7,PALO.SORT(„$E$2"))}

Differently than when sorting with "Alias" as criterion, $B$7 could be omitted here.

Example D9:D25 with the Subset Editor:
The argument **Reverse** is used to reverse the sorting. The reversal of a simple sort order, e.g. a sort by text for all elements (Reverse = 2) is always possible. If, however, the sorting is to be reversed on a particular element level, a level structure has to be generated as well. This is achieved by entering 1 or 2 for the argument "Whole".

### Hierarchical presentation with alphabetical sorting on every level:

E9:E38: `=PALO.SUBSET($A$1,$C$2,$C$5,,PALO.SORT($E$2,$E$5))`

### Hierarchical presentation with parent below the children and alphabetical sorting on every level:

G9:G38: `=PALO.SUBSET($A$1,$C$2,$C$5,,PALO.SORT($E$2,$E$5,,,,$G$2))`

### Pure alphabetical sorting:

I9:I38: `=PALO.SUBSET($A$1,$C$2,$C$5,,PALO.SORT($I$5))`

### Reversal of alphabetical sorting:

K9:K38: `=PALO.SUBSET($A$1,$C$2,$C$5,,PALO.SORT($I$5,,,,$K$2))`

### Column M equal to column E.

### Hierarchical presentation and reversal of the alphabetical sorting on every level:

O9:O38: `=PALO.SUBSET($A$1,$C$2,$C$5,,PALO.SORT($M$2,$M$5,,,,$O$2))`

---

This handbook has been personalized for NA - ohneanmeldung
Example G9:G38 with the Subset Editor:

<table>
<thead>
<tr>
<th>Database</th>
<th>Dimension</th>
<th>Layout</th>
<th>Optional Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demo</td>
<td>Products</td>
<td>Show parents below children.</td>
<td>Show children of hidden elements</td>
</tr>
</tbody>
</table>

Subset Sorting

- **Sorting behaviour**
  - **By Definition**
  - **By Element name**
  - **By Alias**
  - **By Value**
  - **By Attribute**

- **Reverse Order**
  - **Reverse**

- **Type Limitation**
  - **All Elements**
  - **Base Elements Only**
  - **Consolidated Elements Only**

Note that "Reverse" is not selected. This reverse-argument is entered by the layout "Parents below the children" in the register-card **General**.

**Important note in the register-card Sort:**
Not all arguments of `PALO.SORT()` can be selected in the register-card "Sort" by clicking on them, but some of them can be found directly in the register-card "General" (with duplicates) or they are generated by a combination of the settings in "General" and "Sort" (see previous example).
13 Creation of Excel Pivot Tables

13.1 Preliminary notes

ODBO (OLE DB OLAP) is a Microsoft published specification and an industry standard for multidimensional data processing. The Palo ODBO provider is installed with the normal Palo-Setup. It allows the creation of pivot tables using the data from a Palo database.

In order to be accessible through the Palo ODBO provider a Palo database must be conform with the following preconditions:

1. Each dimension must have one top element and only one.
2. Parallel hierarchies are not allowed (for each element one and only one parent is allowed).
3. It is possible to define various measures and time dimensions for a database. However, for MDX access and the Pivot-Table integration, each Cube has to be assigned with exactly one Measures and one Time dimension. The names of this dimensions are arbitrary.

This can be done in the Modeller of the Palo Excel AddIn:

Open the Modeller, and click on the cube for which you want to define a Measures- or Time-dimension. The dimension contained in the cube are expanded. Now right-click the dimension which you want to assign the "Measures" function and select "Treat as measures dimension (MDX)" from the context menu.

This dimension now will be used as Measures dimension in Excel Pivot-Table connections. In the same way, a time dimension can be defined.

4. The following structural constraints for the time dimension.
If there is a time dimension in the cube, then this dimension must have a structure like: 2005_H1_Q1_M12_W4_D1. H stands for half year, Q for quarter, M for month W for week and D for day. Any position in this structure is optional. One can have 2005_D1, or 2005_M2_D15, for example.

Two detailed examples:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>AllYears</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2005</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>2006</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>2007</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>2008</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>2009</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>2010</td>
<td>1</td>
</tr>
<tr>
<td>8 C</td>
<td>2005</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2005_H1</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>2005_H2</td>
<td>1</td>
</tr>
<tr>
<td>11 C</td>
<td>2005_H1</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>2005_H1_Q1</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>2005_H1_Q2</td>
<td>1</td>
</tr>
<tr>
<td>14 C</td>
<td>2005_H1_Q1</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>2005_H1_Q1_M1</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>2005_H1_Q1_M2</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>2005_H1_Q1_M3</td>
<td>1</td>
</tr>
<tr>
<td>18 N</td>
<td>2005_H1_Q1_M4</td>
<td></td>
</tr>
<tr>
<td>19 N</td>
<td>2005_H1_Q1_M5</td>
<td></td>
</tr>
<tr>
<td>20 N</td>
<td>2005_H1_Q1_M6</td>
<td></td>
</tr>
<tr>
<td>21 C</td>
<td>2005_H1_Q2</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>2005_H1_Q2_M4</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>2005_H1_Q2_M5</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>2005_H1_Q2_M6</td>
<td>1</td>
</tr>
<tr>
<td>25 N</td>
<td>2005_H1_Q2_M4</td>
<td></td>
</tr>
<tr>
<td>26 N</td>
<td>2005_H1_Q2_M5</td>
<td></td>
</tr>
<tr>
<td>27 N</td>
<td>2005_H1_Q2_M6</td>
<td></td>
</tr>
<tr>
<td>28 C</td>
<td>2005_H2</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>2005_H2_Q3</td>
<td>1</td>
</tr>
<tr>
<td>30</td>
<td>2005_H2_Q4</td>
<td>1</td>
</tr>
<tr>
<td>31</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>33</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The suffix for H/Q/M/W/D is not bound to a particular number. However the suffix should be consistent with the context. For example, in a structure like 2007_M1_D1, the D element (day) can take any value between 1 and 31 (day of month). In a structure like 2007_M1_W2_D1 the element D can take any value between 1 and 7 (day of week).
### 13.2 Needed Address Changes for the ODBO Connection

There are three possible installation scenarios:

1. Palo Client, Palo Server and Palo ODBO Provider on the same machine.

<table>
<thead>
<tr>
<th>The configuration file for the Palo ODBO Provider is stored in the Palo folder and has four relevant items:</th>
</tr>
</thead>
<tbody>
<tr>
<td>config.ini</td>
</tr>
<tr>
<td>default-Path: C:\Program Files\Jedox\Palo</td>
</tr>
</tbody>
</table>

| PalopAddress 127.0.0.1 |
| MDXAddress 127.0.0.1 |
| PaloPort 7777 |
| MDXPort 4242 |

**PaloAddress** represents the IP address the provider expects to find a Palo Server.

**MDXAddress** represents the IP address the provider is bound to. This item is especially relevant in the case of a multi-IP machine. The IP used here will be the IP the clients use to connect to the provider.

**PaloPort** represents the port used to connect to the Palo Server.

**MDXPort** represents the port the clients use to connect to the provider.

| palo.ini default-Path: C:\Program Files\Jedox\Palo\data |
| cache-barrier 5000 |
| user-login |
| verbose info |
| fake-session |

log "C:\Programme\Jedox\Palo\Log\PaloServer.log"

template-directory "C:\Programme\Jedox\Palo\Api"

**13.2.1 All in One**

This is the default installation. Both the Palo Server and the Palo ODBO provider are on the same machine. Connections to the data are executed from the same machine.

**Here is no need to change ini-files.**
13.2.2 Palo Server on Machine 2

In this configuration the Palo Server is on a remote machine. The machine that connects to the data has the Palo ODBO provider locally:

In this case both the Palo Server and the Palo ODBO Provider requires configuration.

<table>
<thead>
<tr>
<th>The Palo Server:</th>
<th>The Palo ODBO Provider:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edit the Palo.ini file and look for a construct like: http localhost 7777 Change here localhost to the IP of the Machine 2.</td>
<td>Edit the config.ini file and look for a construct like: PaloAddress 127.0.0.1 Change here 127.0.0.1 to the IP of the Machine 2.</td>
</tr>
<tr>
<td>For Example: http 80.72.130.210 7777</td>
<td>For Example: PaloAddress 80.72.130.210</td>
</tr>
</tbody>
</table>

Restart both the local machine and the server for the changes to commit (Alternatively you can restart only the relevant services: Palo and MDX).

13.2.3 Palo Server and Palo ODBO Provider on Machine 2

Here the Palo Server and the Palo ODBO Provider are on the remote machine. However, in order to access the data, the machine used for the client has to have installed a second copy of the Palo ODBO Provider:

In this case, the provider will be split between a thin client interface on the client machine and the main provider working on the server machine. The Palo Client get the IP address of Machine 2 while connecting with Palo Server (chapter 4 and 5). Therefore only the Palo ODBO Provider on Machine 2 needs changes in the config.ini.

| Edit the config.ini file and look for a construct like: MDXAddress 127.0.0.1 Change here the address 127.0.0.1 to the IP of the Machine 2. | For example: MDXAddress 80.72.130.210 |

Restart "Jedox MDX Interpreter" under "Start – Control Panel – Administrative Tools – Services" or restart Machine 2 to commit these changes.
13.3 Using Palo ODBO Provider with Excel 2003

The following steps describe how you can create Pivot Tables in Microsoft Excel 2003 using the Palo ODBO Provider.

Select “Data/PivotTable and PivotChart Report…” from Excel menu:

Select External data source:
Click on Get Data:

Select Tab "OLAP Cubes" from Choose Data Source dialog, then select <New Data Source> and click OK button:

Fill in the fields in the “Create New Data Source” dialog, similar to this one. The name for the data source is free. Make sure you choose Palo ODBO Provider, then click on Connect… button:
After clicking "Connect" you have to enter the server address, user name, password and the name of the source database:

Please enter at "Server Address" the IP-Address of the machine with the "main" Palo ODBO Provider! (look chapter 3.2 and 3.3): This is "127.0.0.1" if the "main" Palo ODBO Provider is on the same machine and the IP-address of machine 2, if there is the "main" Palo ODBO Provider.

**Note: Specify the location of the data means here location of the "main" Palo ODBO Provider!** (After the ini-changes of chapter 3.2 or 3.3 the "main" Palo ODBO Provider knows the location of the Palo Server with the data).

Then enter username and password (standard password for user admin is admin, make sure you change it accordingly). Choose the database to use and Test the connection.

Click on the button "Test Connection" and you receive a message:

If this is successful, you can click OK.
Fill in the missing fields in Create New Data Source dialog:

If you are also mark "Save my user-ID and password in the data source definition" you will no longer ask for it. You will get the following dialog:

When prompted to save password choose "Yes", then click OK:
So you have now immediate access to the stored connection
"Choose Data Source" dialog should now look like this:
Click OK and then click on Finish in PivotTable Wizard dialog:

Confirm with "NEXT".

Now you will be asked where you want to display the report:

If you haven't mark "Save my user-ID and password in the data source definition" you have to enter the user data once more:
With the input of data and click on "OK" is a blank pivot table.

Here you can draw the entries from the PivotTable Field List into the Page Fields, Column Fields and Row Fields.

You will then receive the corresponding PivotTable view.

Below is an example of such an evaluation:
Here you see a comparison between the classical Palo view and the pivot-table view:

![Excel Pivot Table Example](image)

After changing data in the database, it is necessary to update the screen to correctly display the correct data in the pivot table (via the menu Data or via the context-menu on the right-mouse button):
13.4 Using Palo ODBO Provider with Excel 2007

The following steps describe how you can create Pivot Tables in Microsoft Excel 2007 using the Palo ODBO Provider.

Select “Data/From Other Sources/From Data Connection Wizard” from Ribbon:

Select Other/Advanced and click “Next”:

The Data Connection Wizard is displayed, allowing you to choose the data source to connect to.
Choose Palo ODBO Provider from the list and click "Next":

Please enter at "Server Address" the IP-Address of the machine with the "main" Palo ODBO Provider! (look chapter 3.2 and 3.3): This is "127.0.0.1" if the "main" Palo ODBO Provider is on the same machine and the IP-address of machine 2, if there is the "main" Palo ODBO Provider.

**Note:** Specify the location of the data means here location of the "main" Palo ODBO Provider! (After the ini-changes of chapter 3.2 or 3.3 the "main" Palo ODBO Provider knows the location of the Palo Server with the data).

Then enter username and password (standard password for user admin is admin, make sure you change it accordingly). Choose the database to use and test the connection. If this is successful, you can click OK.

This handbook has been personalized for NA - ohneanmeldung
In the next dialog you can choose which Cube you want to query (if more than one is present):

```
Select Database and Table
Select the Database and Table/Cube which contains the data you want.

Select the database that contains the data you want:
Biler

Connect to a specific cube:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Modified</th>
<th>Created</th>
<th>Type</th>
</tr>
</thead>
</table>
```

In the next step you can enter a file name for the connection data, a description and a friendly name. You have the option to save your password in this file.

```
Data Connection Wizard

Save Data Connection File and Finish
Enter a name and description for your new Data Connection file, and press Finish to save.

File Name:
bilier Orders.odc

Save password in file

Description:
(To help others understand what your data connection points to)

Friendly Name:
Bilier Orders

Search keywords:

Always attempt to use this file to refresh data
Excel Services: Authentication Settings...
```

This handbook has been personalized for NA - ohneanmeldung
Final steps should look like this:

![Excel Pivot Table Creation]

This handbook has been personalized for NA - ohneanmeldung
A finished Pivot table in Excel 2007:
14 Tips

14.1 Tip 1: Creating Database Backups

By default, Palo stores your databases in the folder
C:\Program Files\Jedox\Palo\data
or in the folder
C:\Documents and Settings\All Users\(Shared)Documents\Jedox\Palo\data

In order to copy or rename these databases or to create a backup of them or transfer them to a different computer, please do the following:

<table>
<thead>
<tr>
<th>Step</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Go to the Palo Service Manager icon in the task bar. End the service using the mouse button on the right and clicking “End the Palo service” in the context menu.</td>
<td>![Icon]</td>
</tr>
<tr>
<td>The icon switches to:</td>
<td>![Icon]</td>
</tr>
<tr>
<td>Now go to the directory where the Palo databases are saved. As a default, this is C:\Program Files\Jedox\Palo\data or: C:\Documents and Settings\All Users(Shared)Documents\Jedox\Palo\data.</td>
<td>![Icon]</td>
</tr>
<tr>
<td>Create a copy of the relevant database folder in the same directory (in the same manner as with an ordinary folder).</td>
<td>![Icon]</td>
</tr>
<tr>
<td>Go back to the Palo Service Manager in the task bar.</td>
<td>![Icon]</td>
</tr>
<tr>
<td>Start the service with a right-click and select “Start the Palo service” in the context menu. The icon switches to:</td>
<td>![Icon]</td>
</tr>
<tr>
<td>(Depending on the number and sizes of the databases in the directory …Palo\Data, it will take the PALO-Service more or less time to update its state of service)</td>
<td>![Icon]</td>
</tr>
</tbody>
</table>

Please note: The name of the database folder may not contain any blanks! Otherwise, restarting the Palo Service Manager will not work properly.

14.2 Tip 2: Creating new Elements

You opened the Modeller, created a new dimension and opened it. When you press ENTER the first time, a new element is created. It is highlighted and can be replaced by a suitable name. After you have entered the name, you must press ENTER again to confirm. If you press ENTER twice one time after another:

a) Creation of the new element with the name (e.g., "Smith Corp") is completed, and at the same time

b) A new, additional element is created.
14.3  Tip 3: Display Comments in an Array

You create a comments field. It is longer than 255 characters. How you set this up is described in the "Comments" section where we talk about creating or cancelling an array. This is required to display text information in a continuous form.

Creation of an array: Select the cells in question. Caution: The array is created with the data that are in the cell that you selected first. Pay attention to the sequence. Click the "C" from PALO.DATAC and change this to PALO.DATAT. Conclude this entry by pressing the CTRL+SHIFT+ENTER keys.

You have now generated an array which can be recognized by the curved brackets (braces). You cannot easily make changes in this array, e.g., to the formula.

If you want to cancel the array, proceed as follows:
Select exactly the cell area of the array-function. Then click with the mouse in the Formula Bar and press CTRL+CAPS LOCK+ENTER: The array functionality is turned off.

14.4  Tip 4: Use of Consolidation Factors

You can also use the options, which consolidation factors offer, for the following task: Imagine a report that is structured variably. Besides monthly results, you also want to be able to select quarterly or annual results. The report contains key figures that represent full-time employees. Of course, the number of employees is largely identical each month (permanent workforce). Consequently, the number of employees may not be added up in a quarterly report. Otherwise, a multiple of the workforce would be the result.

On the other hand, you have cost positions, for example. These figures need to accumulate to get a quarterly report (consolidation factor = 1). How can you take this requirement into consideration? Create two different consolidated elements per quarter (e.g., Quarter1 and Quarter1 cum.). You define an element with the factor "1" for each of the three months and total up the cost, for example. The second consolidated element has the same months. But each month is only allocated a share of 1/3 (factor = 0.333). As a result, you obtain the average number of full-time employees for all three months. Now you have the correct number in the sum total.

In the same way, you can now create the other three quarters and a special year element, which is composed of these four quarters with a consolidation factor of 0.25 each. This will allow you to perform all normal calculations using consolidation factors.

14.5  Tip 5: Copying Cell Contents (Copy & Like)

This tip refers to copying using the copy command. Problems sometimes occur with the formula due to spelling of the dimension. In this case, you must ensure that the name matches the spelling of the dimension name exactly. Consequently, write "Datatypes" and not "datatypes", for example.
14.6 Tip 6: Importing Text Files

The basis of this tip is an import using the following pattern (in this example, the critical place is the cell "F1"): 

![Excel Spreadsheet](image)

The value in this cell is not recognized as a number in a few installations and may be the case even if the field is formatted correctly otherwise. In this case, Palo outputs an error message (e.g., "Cannot write text in a numeric cell"). To get around this, enter the formula "=Value(F1)" in the neighboring cell (here: G1), and change the "PALO.SETDATA" function correspondingly.

14.7 Tip 7: Formatting at Import

If you use the Data Import Wizard, data are formatted automatically during the import. The cell format is created when "Continue" is clicked. On the other hand, if you click the "Finish" button, it is ignored. The auto-format in cell A1 sets the other values according to regional characteristics (e.g. date, currency and number format). This can result in problems in dependent forms. To prevent this, set the cell format to the format of the source data or to "String".

14.8 Tip 8: Using PALO.SETDATA and PALO.EADD

You know that you should not use PALO.SETDATA and PALO.EADD together in one worksheet. It is possible to use these functions in different worksheets of one workbook. But the EADD functions must then end as a mandatory condition at "FALSE". Consequently, this procedure is not recommended.

It is better if you process both functions in separate workbooks. It also provides a practical benefit, because you will probably need imports for SETDATA functions more often, especially if you import the same type of data from other sources at regular intervals. On the other hand, you only need the EADD functions for the first setup of elements or when importing new elements.

If you proceed this way, you can import the elements using EADD functions. Then close this workbook (e.g., Import Element.xls). Afterwards open the workbook with the SETDATA functions (e.g., Import Data.xls) and perform the corresponding imports. This prevents a possible conflict between the formulas caused by the interdependencies.
15 Administration of Palo

15.1 Data Backup

We recommend that you backup your databases in a regular basis, depending on the level of use. It is sufficient if, after stopping Palo-Server, you copy the databases to another data carrier. There is one folder for each database – the folder is named after the database.

You find the following files on the server system under:

Windows:

- Databases: \Jedox\Palo\data\
- Log files: \Jedox\Palo\Log\

Linux:

- Databases: /opt/palo/data/
- Log files: /opt/palo/Log/

15.1.1 Database Files

Palo databases consist of 3 different datatypes: csv, log and archived files.

The CSV files contain the data last saved. The values currently in storage are registered in the log files. After stopping Palo-Server, the values will be transferred from the log file into the csv file, whereas the data from the csv file will be carried over to the archived files.

If your database takes up a lot of space on the server, you can delete the archived files from the database directory after stopping Palo-Server. After restarting Palo-Server, the needed archived files will be created anew.

Before deleting these files from your database directory, we recommend that you always do a Backup.

Comma Separated Values (CSV):

The structure of the Palo database is set in database.csv. It is here that cubes, dimensions and elements and their characteristics are described.

For each cube that exists in database.csv (entries under [CUBES]) there is in turn its own csv file in which structure and values are specified further. These files contain the currently saved Palo values.

For each Palo rule there is also a unique csv file which contains the description of the rule.
Excerpt from the Palo Demo database with the following entries in database.csv:

[CUBES]
0,"#_GROUP_CUBE_DATA",0,1,1,0,0,
1,"#_Months",3,2,3,0,0,
2,"#_GROUP_DIMENSION_DATA_Months",0,2,1,0,0,
3,"#_Years",5,4,3,0,0,
4,"#_GROUP_DIMENSION_DATA_Years",0,4,1,0,0,
5,"#_Regions",7,6,3,0,0,
6,"#_GROUP_DIMENSION_DATA_Regions",0,6,1,0,0,
7,"#_Datatypes",9,8,3,0,0,
8,"#_GROUP_DIMENSION_DATA_Datatypes",0,8,1,0,0,
9,"#_Measures",11,10,3,0,0,
10,"#_GROUP_DIMENSION_DATA_Measures",0,10,1,0,0,
11,"#_Products",13,12,3,0,0,
12,"#_GROUP_DIMENSION_DATA_Products",0,12,1,0,0,
13,"Sales",12,6,2,4,8,10,2,1,1,
14,"#_#_CUBE_",14,1,3,0,0,

There is another csv file for each entry.

Short description of the individual cubes:

0,"#_GROUP_CUBE_DATA",0,1,1,0,0,

Contains the dimension cubes and groups and the rights of the groups to the cubes.

1,"#_Months",3,2,3,0,0,
3,"#_Years",5,4,3,0,0,
5,"#_Regions",7,6,3,0,0,
7,"#_Datatypes",9,8,3,0,0,
9,"#_Measures",11,10,3,0,0,
11,"#_Products",13,12,3,0,0,

These six cubes contain the attribute cube for each dimension:

2,"#_GROUP_DIMENSION_DATA_Months",0,2,1,0,0,
4,"#_GROUP_DIMENSION_DATA_Years",0,4,1,0,0,
6,"#_GROUP_DIMENSION_DATA_Regions",0,6,1,0,0,
8,"#_GROUP_DIMENSION_DATA_Datatypes",0,8,1,0,0,
10,"#_GROUP_DIMENSION_DATA_Measures",0,10,1,0,0,
12,"#_GROUP_DIMENSION_DATA_Products",0,12,1,0,0,

These cubes contain the user group and the dimensions. The rights for each element for each group is registered here.

13,"Sales",12,6,2,4,8,10,2,1,1,

Contains the data for the cube 'Sales'.

14,"#_#_CUBE_",14,1,3,0,0,

This cube contains the data for the attribute cube for each cube dimension.
Log Files:
A log file is created for each cube defined in Palo. All values entered in Palo are stored directly in this file. If the database is saved, the value stored in here moves to the respective csv file.

Archived Files:
All changes are written into the archived files. Both value changes as well as structural changes are logged together with the name of the user. In case the database corrupts, Palo-Server can reinstate it with the archived files.

Archived files may be deleted regularly (after a backup), to keep the data quantity in the server system low. A Palo Services restart will create these files anew.

15.1.2 PaloXIAddin.log und PaloServer.log
A log file each exists for the Palo Excel-Addin and the Palo-Server (PaloXIAddin.log und PaloServer.log). The PaloServer.log file can grow in size relatively quickly. The detail level of logging can be set in palo.ini. The different detail levels for this log are: error, warning, info, trace and debug. The default setting is "info".

A short description of the various levels:
- **error**: Output of fatal errors:
  If this occurs, Palo-Server should be stopped (the Server may stop itself) to correct the error. It is not advisable to ignore the error as it may lead to database corruption. Example: no storage space left on the disc.
- **warning**: Output of warnings:
  This error as well should be examined. However, it should not generally lead to a corrupt DB.
- **info**: Output of general notes concerning the mode of operation
- **trace & debug**: This output is very detailed and serves the Palo development

After changes in palo.ini have been made, Palo-Server has to be restarted to make the changes effective. Both of these log files can also be deleted or saved after stopping Palo-Server. Restarting Palo will create these files anew.

15.2 Entries in the Windows Registry of the Client
Starting with Palo 2.0, Palo connections to the servers and the paths to the import/export files are registered in the Registry. If you enter regedit on Windows>Start>Run and confirm the command, the registry editor will open.

For each server registered via the Palo Wizard, an entry exists showing the parameters: connected (display of the status of the server connection), hostname, password, port and username. These entries, connected, hostname, port and username, can be read directly. The password is encrypted and should not be changed. After changes have been made, Excel has to be closed and the Palo-Server restarted to make the changes effective.

The paths to the Import-(CSV) and Export-files (CSV2) can be find under HKEY_CURRENT_USER\Software\Jedox\Palo\XIAddin\MRU (Most recently Used).

15.3 Starting Palo Service via the Console
Palo Service can be started directly via the Windows Console and Net Start. The commands are:

```
c:\net start paloserverservice
```

```
c:\net stop paloserverservice
```
15.4 Cache Settings of the Server

Ideally, Palo-Server is run on a target system with a lot of memory and a fast processor as Palo loads the data into RAM when starting. The memory capacity depends on the size of the databases and the calculation of consolidated values.

15.4.1 Cube-Caches

The base cells of the cubes that have been loaded by Palo are always stored inside the main memory of the machine the Server is running on. The aggregated cells, however, have to be calculated whenever their value is requested. The cache improves on this situation by storing previously requested aggregations.

The cache is divided into logical cube-caches. There is one cube-cache for every cube.

To set the values for any of the parameters, please use the `palo.ini` file (palo -? at the command line shows the parameters for the command line input, if for some reason the palo.ini is not used).

15.4.2 Parameters in the palo.ini

There are four parameters that can be changed to improve the Server Side Cache performance of Palo.

- **cache-barrier:** The higher this value, the smaller the probability that an aggregated cell is cached. If the number of base elements of an aggregated cell exceeds the value of `cache_barrier`, Palo stores the calculated cell value inside the cache. Otherwise it does not. The general idea behind this parameter is that consolidations with few base elements are less expensive to calculate than those with many. Therefore, the performance hit is greater for aggregations whose values are computed from many cells. If the cache is used to store the expensive cell values only, the efficiency is higher than when using it to store any kind of aggregation. However, the parameter should not be set to a value that is too high. This will result in the cache not being used to its full capacity.

- **cache-size:** This parameter determines the total number of values the cache is able to hold. If this number is reached, half of every cube-cache will be emptied. Here, the strategy is to keep those values that have been least recently used. The recommended size of this value depends on many factors. These include total amount of RAM and number of aggregations inside the cubes. The default value is 100MB. Note that this is not a per-cube but a total limit.

- **clear-cache:** This parameter determines the number of invalidations after which a cube-cache is cleared completely. An invalidation is the change or deletion of a cell-value of a cell of any kind. Invalidations are counted on a per-cube basis. If the number of invalidations exceeds this parameter, the cache of the affected cube is cleared completely. Note that the change of an aggregation changes many base cells (splashing).

- **clear-cache-cells:** Determines the number of changed based cells (a special kind of invalidation) after which a cube-cache is cleared completely. Works like clear cache, except that only base-cells are considered.

**Example**

Here, an example for the effect of the clear_cache and clear_cache_cells parameters is given. Let clear_cache be set to 5000 and let clear_cache_cells be set to 100.

Now, if an aggregation with 200 base cells is changed, this is below the limit set by clear_cache but above the clear_cache_cells limit. Therefore, the cube-cache of the cube the aggregation belongs to will be cleared completely.

Now, if clear_cache has value 5000 and clear_cache_cells has value 10000, it will take 25 changes to the above aggregation to reach the clear_cache limit.
15.5 Multithreading

Multithreading means that Processes and tasks can be distributed on multiple processors / cores. This ability of the Palo OLAP Server is available only in the Enterprise version. The settings can be found at the end of the file palo.ini.sample (see chapter 2.5).

15.6 Formatting the Display of Palo Data in Excel

The installation directory of Palo contains a file named "palostyles.xls", which enables you to control the formatting of Palo data that is retrieved via the "Paste View". This may be necessary, to control the display of decimal places of numbers, for example.

Before you open the template in Excel, we recommend that you make a backup of the existing file. The formatting is changed via Format > Style > Modify. For the changes to become effective you have to restart Excel.

Example: Changing the data display to zero decimal places:
Open the file palostyles.xls.
Select Menu/Format/Style.../Style name _data,
then click on Modify... and change the number format to zero decimal places.
Save the file palostyles.xls, close Excel and then start it again.
16 Overview of Palo Functions

16.1 Classification of Functions

The Palo functions can be classified as follows:

1. According to how they are called:
   a) Via the function wizard in Excel (Paste/Function – Category: Palo)
   b) Direct entry

2. According to their purpose:
   a) Data input, data output
   b) Data management, information on and overview of structures
   c) Calculations and splits

3. According to their area of use:
   a) Table functions
   b) Array functions
   c) Splitting functions

You already learned the most important table functions in the previous chapters. Simply put, they serve data input and output. Consequently, you will use them again and again when you work with Palo.

In addition to the table functions, there are also numerous functions that primarily serve data management. For example, they enable you to check data structures, determine dependencies and similar matters. These functions are described below, one example each.

Array functions help to control and analyze structures. You can call both table and array functions via the function wizard of Excel. The required parameters are described in the table below. You can find these above the list of functions.

The pure splitting functions can only be used via direct entry. This includes the functions on the topics of splash, copy and like described in the "Working with the Modeller" chapter. "Palo.Setdata" is also one of the functions, which you can use to calculate.

Argument "Empty String": In the current version, various functions have been extended to accept the argument "Empty String". If you enter "TRUE" as actual argument here, an error message, if it occurs, will not be displayed – the cell will be "left empty"!

This handbook has been personalized for NA - ohneanmeldung
## 16.2 Input Parameters

You can find parameters and their description in the table below.

<table>
<thead>
<tr>
<th>Name</th>
<th>Possible entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server/Database</td>
<td>A text (Name) such as &quot;localhost/Demo&quot; or a coordinate such as A1</td>
</tr>
<tr>
<td>Cube:</td>
<td>A text (Name) such as &quot;Sales&quot; or a coordinate such as A2</td>
</tr>
<tr>
<td>Dimension</td>
<td>A text (Name) such as &quot;Europe&quot; or a coordinate such as A4</td>
</tr>
<tr>
<td>Element/consolidated element</td>
<td>A text (Name) such as &quot;Variance&quot; or a coordinate such as B8</td>
</tr>
<tr>
<td>Coordinate</td>
<td>Can be an element, for example:</td>
</tr>
<tr>
<td>Type</td>
<td>&quot;n&quot; for numeric, &quot;s&quot; for String/Text and &quot;c&quot; for consolidated</td>
</tr>
<tr>
<td>Weighting factor</td>
<td>A number</td>
</tr>
<tr>
<td>Delete</td>
<td>True or false (true if existing entries should be deleted)</td>
</tr>
<tr>
<td>Position of base element</td>
<td>A number</td>
</tr>
<tr>
<td>Position</td>
<td>A number</td>
</tr>
<tr>
<td>Number of consolidated element</td>
<td>A number</td>
</tr>
<tr>
<td>Old name, new name</td>
<td>Two coordinates such as A1,B1. <strong>Note:</strong> the names to be changed are in the coordinates.</td>
</tr>
<tr>
<td>Position of the sibling element</td>
<td>A number</td>
</tr>
<tr>
<td>Value</td>
<td>The coordinate that holds the value to be written. <strong>Note:</strong> Instead of a coordinate such as A1, you can also use a formula such as A1*B1. This is explained in the push rules.</td>
</tr>
<tr>
<td>Splash</td>
<td>True or false (true = write in consolidated element)</td>
</tr>
<tr>
<td>Error</td>
<td>A coordinate, e.g., B10. This refers to the cell, in which the appropriate Palo.Setdata function is executed.</td>
</tr>
<tr>
<td>Storage location (path)</td>
<td>A path such as C:\Documents and Setting\Own Files\Error log.txt. Then the faulty data records of an import are saved in the &quot;Error log.txt&quot; file.</td>
</tr>
</tbody>
</table>

**Notes:**

You might have to replace the comma (,) with a semicolon (,;) and the apostrophe (’) with quotation marks (""). This depends on the country settings.

If an argument remains blank in a formula, two quotation marks (""") are normally set to generate a blank string.
In addition, examples are given in the individual functions. In the screen below, for instance, you see that you can enter "localhost/Demo" or "Sales". Alternatively, you can use the coordinates A1, A2, etc.

### 16.3 Table Functions

**PALO.DATA(Server/Database, Cube, Coordinate1, Coordinate2, ..., Coordinate16)**
Reads the value of the described data cell from the cube or writes it back, e.g.:
PALO.DATA("localhost/Demo","Sales","Desktop L","Germany","Oct",2002,"Actual","Units")

**PALO.DATAC(Server/Database, Cube, Coordinate1, Coordinate2, ..., Coordinate16)**
Reads the value of the described data cell from the cube or writes it back, e.g.:
PALO.DATAC("localhost/Demo","Sales","Desktop L","Germany","Oct",2002,"Actual","Units")

Contrary to PALO.DATA, all DATAC formulas of a worksheet are calculated at once and it is consequently faster as PALO.DATA. But other Palo formulas do not work if they refer to the result of a PALO.DATAC function.

**PALO.DATAT(Server/Database, Cube, Coordinate1, Coordinate2, ..., Coordinate16)**
Reads the value of the descriptive text cell from the cube or returns it, e.g.:
PALO.DATAT("localhost/Demo","Market","Comment","Year","Units","All Products","2006","Total")

This function enables display of more than 255 characters if they are entered in an array function. In this case, the continuation of the text is displayed in the next cell of the array.

**PALO.DATAV(Server/Database, Cube, Coordinate1, Coordinate2, ..., Coordinate16)**
Returns the value of a data cell from a cube when you use an Excel array formula.

It is similar to PALO.DATA and PALO.DATAC, but the function works even faster because a single array formula is generated for the complete area. Use of PALO.DATAV is only possible in related cell areas.
PALO.DATAX(Server/Database,Cube, coordinate1, coordinate2,... coordinate16,)
Returns the value of the described element from the cube, if the corresponding cell is on the currently active work sheet.

Note: This function corresponds to the PALO.DATA-function. However, the computation is limited to the currently active table. This speeds up the computation, because it is not necessary any more to compute the whole workbook.
If however, you switch to a different work sheet and later come back to the work sheet which contains such PALO.DATAX-formulas, it may be necessary to manually trigger a re-computation of the work sheet. This will always be necessary, if calling the work sheet returns "#NV", instead of a value.

PALO.EADD(Server/Database,Dimension,Type,Element,Consolidated Element,Weighting factor,Delete)
Adds the described dimension element or structures to a database.
Example 1: PALO.EADD("localhost/Demo","Years","n","2010","",1,FALSE) – consolidated element = BLANK here.
Example 2: PALO.EADD("localhost/Demo","Months","c","Qtr.1","Year",1,FALSE)
Possible parameters values are 0 (or FALSE), 1 (or TRUE) and 2
0: Removes before importing no elements of the dimension
1: Removes before importing all elements of the existing dimension
2: Removes before importing all existing elements of the C-dimension, basic elements are not deleted.

Note 1: Make sure that all EADD functions referring to the same dimension use identical parameters (true or false). Otherwise, Palo would execute that parameter for the complete dimension that happens to be found first. Reason: the sequence, in which Excel formulas are processed, cannot be determined. During an import the command delete will only be executed one time with the first PALO.EADD () and then it will be ignored.

Note 2: The function PALO.EADD can only be used in connection with an import, because the import wizard recalculates the database. Compare the description in the "Importing and Exporting Data" chapter.

PALO.ECHILD(Server/Database,Dimension,Consolidated Element,Position of base element)
Returns the name of the described base element.
Example: PALO.ECHILD("localhost/Demo","Regions","South",3) returns "Spain"

PALO.ECHILDCOUNT(Server/Database,Dimension,Consolidated Element)
Returns the number of base elements in a described consolidated element.
Example: PALO.ECHILDCOUNT("localhost/Demo","Regions","South") returns "3"

PALO.ECOUNT(Server/Database,Dimension)
Returns the total number of elements in the described dimension.
Example: PALO.ECOUNT("localhost/Demo","Years") returns "8"

PALO.EDELETE(Server/Database,Dimension,Element)
Deletes the described element
Example: PALO.EDELETE("localhost/Demo","Products","Subnote XK"

Note: The function PALO.EDELETE can only be used in connection with an import, because the deletion is controlled via the import. Compare the example in the "Importing and Exporting Data" chapter.

PALO.EFIRST(Server/Database,Dimension)
Returns the element in the described dimension.
Example: PALO.EFIRST("localhost/Demo","Regions") returns "Germany"

PALO.EINDENT(Server/Database,Dimension,Element)
Returns the hierarchy level of an element within the dimension.
Example: PALO.EINDENT("localhost/Demo","Regions","Italy") returns "3"
PALO.EINDEX(Server/Database,Dimension,Element)
Returns the described element in the dimension.
Example: PALO.EINDEX(“localhost/Demo”,“Regions”,“France”) returns “2”.

PALO.EISCHILD(Server/Database,Dimension,Consolidated Element,Element)
Checks whether a consolidated element contains the described element.
Example: PALO.EISCHILD(“localhost/Demo”,“Regions”,“West”,“Germany”) returns TRUE.

PALO.ELEVEL(Server/Database,Dimension,Element)
Returns the number of hierarchy levels within the dimension, which follow after the element.
Example: PALO.ELEVEL(“localhost/Demo”,“Regions”,“Germany”) returns “0”.
Example: PALO.ELEVEL(“localhost/Demo”,“Regions”,“West”) returns “1”.

PALO.ENAME(Server/Database,Dimension,Position)
Returns the name of the element of a specific position. (The first position is 1)
Example: PALO.ENAME(“localhost/Demo”,“Regions”,2) returns “France”.

PALO.ENEXT(Server/Database,Dimension,Element)
Returns the name of the following element within the dimension.
Example: PALO.ENEXT(“localhost/Demo”,“Regions”,“Italy”) returns “Portugal”.

PALO.EPARENT(Server/Database,Dimension,Element,Number of Consolidated Element)
Returns the name of the n th parent of the specified element.
Example: PALO.EPARENT(“localhost/Demo”,“Regions”,“Italy”,1) returns “South”.

Note: If an element occurs only in one consolidation, only the parameter "number" = "1" is possible. If an
element occurs in several consolidations, the "Number" determines which consolidated element is
returned.

Example: PALO.EPARENT(“localhost/Demo”,“Regions”,“Italy”,2) returns “New Group”, because this has
been created and “Italy” is consolidated in it.

PALO.EPARENTCOUNT(Server/Database,Dimension,Element)
Returns the number of consolidated elements, in which the described element is contained.
Example 1: PALO.EPARENTCOUNT(“localhost/Demo”,“Regions”,“Italy”) returns “1”.
Example 2: PALO.EPARENT(“localhost/Demo”,“Regions”,“Italy”,2) returns “2” if there is a consolidated
element “New Group” and “Italy” is consolidated in it.

PALO.EPREV(Server/Database,Dimension,Element)
Returns the name of the preceding base element.
Example: PALO.EPREV(“localhost/Demo”,“Regions”,“Italy”) returns “Austria”.

PALO.ERENAME(Server/Database,Dimension,Old Name,New Name)
Renames the described element.
Example: PALO.ERENAME(“localhost/Demo”,“Products”,“Notebook TT”,“Notebook TS”
Renames the element “Notebook TT” to “Notebook TS”.

Note: The function PALO.ERENAME can only be used in connection with an import, because the
renaming is controlled via the import. Compare the example in the “Importing and Exporting Data”
chapter.

PALO.ERROR_LOG(Error,Storage Location,Value,Cube,Coordinate1, Coordinate2,…, Coordinate16)
Example: PALO.ERROR_LOG (F3,”c:\Error.txt”,A1,B1,C1,D1,E1,F1)
Writes the data records of an import file, which does not adhere to import specifications, into an error file.

Note: The function PALO.SETDATA_ERROR can only be used in connection with an import. Compare the
example in the “Importing and Exporting Data” chapter.
PALO.ESIBLING(Server/Database, Dimension, Element, Position of Sibling Element)
Returns a sibling element of the specified element. The sibling, which is to be returned is specified with a position number. The element itself has the position 0, the sibling in the hierarchy view found directly below has the position 1, etc., the sibling found directly above has the position -1, etc.

Example: PALO.ESIBLING("localhost/Demo", "Regions", "Portugal", 1) returns "Spain".
Palo.ESIBLING( "localhost/Demo", "Regions", "Portugal", -1) returns "Italy".

Note: This function can also be used to verify whether an element exists in a dimension. Therefore the name of the element and the position "0" is entered. If the element exists, the name of the element is returned, if not, "# Name?" is returned.

Example: PALO.ESIBLING("localhost/Demo","Regions","Italy",0) returns "Italy".
Palo.ESIBLING( "localhost/Demo", "Regions", "Greece", 0) returns "#Name?".

PALO.ETOPLEVEL(Server/Database, Dimension)
Returns the number of hierarchy levels in a dimension.
Example: PALO.ETOPLEVEL( "localhost/Demo", "Regions" ) returns "2".

PALO.ETYPE(Server/Database, Dimension, Element)
Returns the type of the described element (numeric, text or consolidated). The return is in English (numeric, string or consolidated).
Example: PALO.ETYPE("localhost/Demo","Regions","Italy") returns "numeric".

PALO.EUPDATE(Server/Database, Dimension, Element, Elementtype, {Element, Consolidationfactor, Element, Consolidationfactor, ...})
Updates an element after the specified data.
To demonstrate this function here a nonsense example:
Palo.EUPDATE=("localhost/demo","Months","Jan","C",{"Feb",1,"Jun",2})
After this import command is Jan a C-element, consolidated with 1xFeb and 2xJun.

A correction of the last nonsense example would be the following import:
=PALO.EUPDATE("localhost/Demo","Months","Jan","N",{0,0}).

PALO.EWEIGHT(Server/Database, Dimension, Consolidated Element, Element)
Returns the weighting factor of the described element at consolidation.
Example: PALO.EWEIGHT("localhost/Demo","Regions","West","Germany") returns "1".

PALO.SETDATA(Value,Splash,(Server/Database,Cube,Coordinate1,Coordinate2,.....))
Example 1:
Palo.SETDATA($G$1,FALSE,"localhost/Demo","Market","D$2","C$1","A$1","E$2","Units","B$1)
Example 2:
Palo.SETDATA(B10*C10*$D$7,False,$A$1,$A$2,$A10,$A$4,$A$5,$A$6,$A$7,D$9)

Writes the values of elements into a cube. They are written into the cells specified through the coordinates. Set SPLASH to "True" if you want to write in a consolidated element. Otherwise, set it to "False". The "Value" parameter is a cell of the Excel worksheet, e.g., A1. Alternatively, it can be a formula.

PALO.SETDATAIF(Condition,Value,Splash,Server/Database,Cube,Coordinate1,Coordinate2,....)
If condition is TRUE, this function writes the values of the elements in the cube – same as PALO.SETDATA () - and if the condition is FALSE, nothing happens.

This handbook has been personalized for NA - ohneanmeldung
16.4 Array-Functions

You must absolutely adhere to the following: The array functions require some preparatory steps:

1. You must highlight an empty range in a worksheet. The results are then returned in this area. But you can also define the area somewhat bigger. Cells not required remain blank in the output.
2. When you enter the function, you must close it with the keyboard shortcut CTRL+SHIFT+ENTER.
3. If you call the function via Paste/Function, close it with CTRL+SHIFT+clicking OK using the mouse.

The Procedure is explained using the example of PALO.CUBE_LIST_DIMENSIONS. It is the same for all other functions in principle.

PALO.CUBE_LIST_DIMENSIONS (Server/Database,Cube)
Example: PALO.CUBE_LIST_DIMENSIONS ("localhost/Demo","Sales")

The function lists the dimensions of a cube in an area you define in advance. Define the corresponding area of an Excel worksheet (e.g., B3:B11).

Then enter this function in the first cell of the area. Conclude the entry by pressing the CTRL+SHIFT+ENTER keys. You are already familiar with this from the comment fields. If you call the function via Paste/Function, you must close it with CTRL+SHIFT+clicking OK using the mouse.

Then the following screen appears:

PALO.DATABASE_LIST_CUBES(Server/Database)
Example: PALO.DATABASE_LIST_CUBES("localhost/Demo")
The function lists the cubes of a database in an area you define in advance.

PALO.DATABASE_LIST_DIMENSIONS(Server/Database)
Example: PALO.DATABASE_LIST_DIMENSIONS("localhost/demo")
The function lists the dimensions of a database in an area you define in advance.

PALO.DIMENSION_LIST_CUBES(Server/Database,Dimension)
Example: PALO.DIMENSION_LIST_CUBES("localhost/Demo","Months")
The function lists, in an area you define in advance, in which cubes a dimension is contained.
PALO.DIMENSION_LIST_ELEMENTS(Server/Database, Dimension)
Example: PALO.DIMENSION_LIST_ELEMENTS("localhost/demo","months")
The function lists, in an area you define in advance, which elements are contained in a dimension.

PALO.EALIAS(Server/Database, Dimension, Attribute, Alias, Index)
The function lists, in an area you define in advance, all elements with the matching alias.
Example: In the database "Demo" was created the attribute "Color" for "Products" and nine products have received the attribute "black".
Palo.EALIAS("localhost/Demo","Products","Color","black") displays this nine products,
Palo.EALIAS("localhost/Demo","Products","Color","black", 3) displays the 3rd of the nine products,
Palo.EALIAS("localhost/Demo","Products","Color","black", 10) displays again the nine products, because the index 10 does not exist.

PALO.ELEMENT_LIST_ANCESTORS(Server/ Database, Dimension, Element, Empty String)
Example: PALO.ELEMENT_LIST_ANCESTORS("localhost/Demo","Months","Jan")
The function lists in a predefined area all ancestors of the given element.

PALO.ELEMENT_LIST_CHILDREN (Server/ Database, Dimension, Consolidated Element)
Example: PALO.DIMENSION_LIST_CHILDREN("localhost/Demo","Regions","West")
The function lists in a predefined area all child elements of a consolidated element.

PALO.ELEMENT_LIST_DESCENDANTS(Server/ Database, Dimension, Element, Empty String)
Example: PALO.ELEMENT_LIST_DESCENDANTS("localhost/Demo","Products","Monitors")
The function lists in a predefined area all children and grandchildren of an element.

PALO.ELEMENT_LIST_PARENTS(Server/ Database, Dimension, Element)
Example: PALO.ELEMENT_LIST_PARENTS("localhost/Demo","Months","Jan")
The function lists in a predefined area all parents of an element.

PALO.ELEMENT_LIST_SIBLINGS (Server/ Database, Dimension, Element)
The function lists in a predefined area all siblings of an element.

PALO.ROOT_LIST_DATABASES(Server)
Example: PALO.ROOT_LIST_DATABASES("localhost")
The function lists the databases that exist on the server, in an area you define in advance. Note: Even if "Server/Database" is written in the formula editor, entry of the server suffices.

16.5 Filter Functions
The filter functions PALO.HFILTER(), PALO.TFILTER(), PALO.PICKLIST(), PALO.AFILTER(), PALO.DFILTER() and PALO.SORT() are sub-functions of PALO.SUBSET(). They do not work independently, but need PALO.SUBSET() as the basic function. In addition, the function PALO.SUBCUBE() is required when using PALO.DFILTER(). The comprehensive description of the filter functions, see the separate chapter "Filter Functions with Subsets".
16.6 Splitting Functions

The splash functions such as COPY and LIKE belong to the splitting functions. Contrary to the previously cited groups, they cannot be controlled via the function wizard from Excel (Paste/Function).

Splash Parameters #

Example: #250

Equally splits the entered value among all subsequent base elements." If the consolidated element is "0" at entry, splitting is uniform. Otherwise, splitting is according to existing shares.

Splash Parameters !!

Example: !!10

Two exclamation marks followed by a value add this value to the existing values of all subsequent base elements. If you assume that a consolidated element has the value 120 and is composed of 19 base elements, then entry of "!!10" results in 120+10*19 =310.

Splash Parameters # … %

Example: #10%

This parameter changes the existing values of a consolidated element by the entered percentage. As a result, 570 becomes with #10% = 627.

Splash Parameters !

Example: !10

You overwrite all following base elements with this value. With "!10", the value 341 is replaced in 19 base elements *10 = 190.

Copy

The COPY function copies the contents and splitting structures of a consolidated element into another one. This can be meaningful, for example, if the actual data for one year should be used for planning the subsequent year.

The formula is: Copy Element,Year. If you want to enter the description more precisely, you can also use Copy Dimension:Element,Year.

Example: Copy Actual,2006 or alternatively: Copy Datatypes:Actual,2006

Like

The Like command contains the COPY command. It uses the shares of the base elements to split the values placed before LIKE. As a result, a different value estimate is possible than that in COPY. The formula is: NUMBER Like Element,Year

Example: 12000000 Like Actual,2006

Please note that it is not possible to enter negative value as e.g. "-1000 LIKE Budget,2007". An alternative that gets you the same result is to "copy Budget,2007" and to splash afterwards -1000(#-1000).
A big THANK YOU goes to all volunteer translators who have made it possible for us to offer Palo in many different languages:

<table>
<thead>
<tr>
<th>Language</th>
<th>Translator</th>
<th>email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinese</td>
<td>Gaojun Pan</td>
<td><a href="mailto:pgj_27@163.com">pgj_27@163.com</a></td>
</tr>
<tr>
<td>Czech</td>
<td>Jan Jonášek</td>
<td><a href="mailto:jjonasek@sefima.cz">jjonasek@sefima.cz</a></td>
</tr>
<tr>
<td>Danish</td>
<td>Bent MØller Madsen</td>
<td></td>
</tr>
<tr>
<td>Dutch</td>
<td>Henk Scholten / BI-Team b.v.</td>
<td><a href="mailto:hscholten@bi-team.com">hscholten@bi-team.com</a></td>
</tr>
<tr>
<td>French</td>
<td>Louis Jaouan</td>
<td><a href="mailto:louis.jaouan@coopanet.com">louis.jaouan@coopanet.com</a></td>
</tr>
<tr>
<td>French</td>
<td>Luc Mornat / BENCHECO</td>
<td><a href="mailto:consulting@bencheco.com">consulting@bencheco.com</a></td>
</tr>
<tr>
<td>Hungarian</td>
<td>Veronika Tatár</td>
<td><a href="mailto:veronika.tatar@vargabor.hu">veronika.tatar@vargabor.hu</a></td>
</tr>
<tr>
<td>Hungarian</td>
<td>Bálint Varga</td>
<td><a href="mailto:balint.varga@vargabor.hu">balint.varga@vargabor.hu</a></td>
</tr>
<tr>
<td>Polish</td>
<td>Lukasz Bulak</td>
<td></td>
</tr>
<tr>
<td>Polish</td>
<td>Zenon Biniek</td>
<td><a href="mailto:zenonb@finus.com.pl">zenonb@finus.com.pl</a></td>
</tr>
<tr>
<td>Portuguese</td>
<td>Elmo Gomes</td>
<td><a href="mailto:elmogomes@dokeo.com.br">elmogomes@dokeo.com.br</a></td>
</tr>
<tr>
<td>Russian</td>
<td>Dmytro Neustroyev</td>
<td><a href="mailto:dmytro.deustroyev@ua.ey.com">dmytro.deustroyev@ua.ey.com</a></td>
</tr>
<tr>
<td>Russian</td>
<td>Sergey Semchenkov</td>
<td><a href="mailto:amfibiis@mail.ru">amfibiis@mail.ru</a></td>
</tr>
<tr>
<td>Serbian</td>
<td>Igor Mijakovic</td>
<td><a href="mailto:smorsmor@gmail.com">smorsmor@gmail.com</a></td>
</tr>
<tr>
<td>Spanish</td>
<td>Mark Dickfeld</td>
<td></td>
</tr>
<tr>
<td>Spanish</td>
<td>Maximino Cares</td>
<td><a href="mailto:mcaresr@gmail.com">mcaresr@gmail.com</a></td>
</tr>
</tbody>
</table>

This handbook has been personalized for NA - ohneanmeldung
18 Index

A
activate the array-functionality · 313
admin · 270, 271
administrator · 24, 262, 280, 285
Alias · 307, 317
allowed characters · 86
argument „Empty String“ · 374
array-function · 62, 129, 132, 317, 319
array-functionality · 131, 313
attribute cube · 101, 108, 110, 119, 120, 134, 136
attribute cubes · 61
attributes · 15, 26, 100, 101, 102, 105, 106, 108, 109, 110,
112, 113, 114, 115, 116, 117, 118, 119, 120, 127,
134, 279
automatic backup · 30
autosave · 30

B
backup · 48, 366
base elements · 65, 95, 103, 143, 145, 149, 150, 151, 152,
153, 158, 159, 160, 183, 220, 377, 382
cell data · 278, 283
characters allowed in cube-names · 86
characters allowed in database-names · 86
characters allowed in dimension-names · 86
characters allowed in element-names · 86
child element · 109
Clear Cube · 162, 164
Clear list · 68
Clear selection · 163
Client-Connection with a Palo-Server · 29
Column titles · 62, 70, 101, 114, 125, 126, 144, 145, 157,
192, 201, 251
command line parameters · 31
comments · 127
Connecting Palo-Client with local Palo-Server · 40
Connecting Palo-Client with Palo-Server in the Network ·
42
Connection with the API · 44
consolidated · 375
consolidated element · 91, 109, 112, 114, 117, 118, 140,
143, 183, 185, 377, 378, 379, 382
consolidation factor · 94, 143, 159, 183, 367
county menu · 93, 96, 97, 98, 100, 126, 127, 171, 234,
235, 366
cordinate · 375
copy · 157, 158, 161, 367, 374, 382
Copy elements with hierarchies into a spreadsheet · 94
create new database · 177
creating cubes · 121
creating databases · 229
cube · 26, 46, 52, 54, 55, 57, 58, 79, 85, 101, 104, 114,
119, 121, 122, 123, 124, 125, 126, 127, 132,
133, 139, 142, 143, 144, 157, 161, 162, 163,
166, 167, 169, 178, 182, 192, 193, 196, 197,
198, 199, 206, 207, 212, 213, 215, 219, 220,
226, 229, 239, 240, 276, 278, 283, 375, 376,
379, 380
Cube #_CONFIGURATION · 277
Cube #_GROUP_CUBE_DATA · 275
Cube #_GROUP_DIMENSION_DATA <Name of
dimension> · 276
Cube #_GROUP_ROLE · 272
Cube #_ROLE_RIGHT_OBJECT · 270
Cube #_USER_GROUP · 273
Cube #_USER_USER_PROPERTIES · 273
Data cubes · 61
Data Import Wizard · 180
database · 11, 18, 26, 31, 32, 45, 46, 48, 52, 57, 58, 79, 83,
85, 95, 121, 124, 124, 132, 139, 142, 148, 166, 169,
172, 176, 178, 180, 185, 189, 190, 191, 193,
211, 223, 224, 225, 226, 227, 228, 229, 230,
239, 266, 276, 366, 375, 376, 377, 378, 379,
380, 381
database source name · 190
deactivates the array-functionality · 313
deinstalling Palo · 50, 224
delete · 45, 113, 125, 126, 127, 161, 183, 186, 188, 200,
223, 224, 226, 227, 228, 251, 285
Deleting a Server-Connection · 43
deleting databases · 227
dimension · 26, 46, 52, 53, 54, 58, 60, 64, 75, 76, 77,
78, 79, 85, 87, 88, 90, 91, 93, 95, 96, 98, 100,
105, 109, 121, 123, 124, 125, 126, 127, 132,
143, 144, 145, 151, 158, 162, 166, 172, 176,
178, 182, 183, 184, 185, 212, 215, 219, 226,
230, 231, 232, 234, 235, 237, 239, 240, 242,
249, 251, 252, 276, 278, 366, 367, 375, 377,
378, 379, 380, 382
Dimension · 379
directories · 28, 29
disable array-functionality · 132
display of hierarchies · 92
Drill History · 27
Drill Palo Rule · 27, 294
Drill Through · 27
DSN · 190, 191
editor · 270, 271, 284
element · 80, 91, 95
element name hide · 277
element type · 94, 95
elements · 26, 60, 69, 74, 75, 76, 77, 78, 81, 85, 93, 96, 97,
100, 103, 105, 107, 109, 114, 117, 118, 124,
127, 132, 133, 142, 143, 166, 172, 176, 182,
183, 184, 185, 186, 188, 192, 207, 213, 219,
232, 233, 234, 235, 236, 242, 243, 244, 245,
278, 366, 367, 375, 377, 378, 379, 381
Empty String · 374
enable the array-functionality · 131
enter data · 183
ersors at import · 215
ETL tools · 214
Excel Pivot Tables · 350
This handbook has been personalized for NA - ohneanmeldung
exporting data - 219

**F**

Filter Functions - 307
Fixed width - 70
Format Painter - 250
Formatting the Display of Palo Data in Excel - 373

**G**

groups - 24, 61, 382
Guess Arguments - 79, 136, 169, 196, 198, 246

**H**

hierarchy - 58, 91, 92, 127
Hierarchy Buttons - 63
hierarchy-level - 316
hierarchy-numbering - 316

**I**

import - 15, 18, 26, 161, 166, 170, 172, 173, 175, 179, 180, 182, 184, 185, 186, 187, 188, 189, 190, 192, 193, 197, 201, 205, 206, 207, 214, 215, 217, 218, 368, 377, 378
importing comments - 175, 176
importing elements - 176
Indent - 70, 307, 315
input parameter - 375
installation - 10, 11, 12, 14, 15, 21, 22, 23, 24, 25, 224, 256, 261, 264
internal loop - 167, 200, 205
Invert selection - 67

**L**

level-selection - 65
like - 157, 159, 183, 367, 374, 382
Linux-Installation - 10
locations of files - 28, 29

**M**

marker - 296
marker arguments - 297
marker definition - 296
marker for expressions - 297
menu Palo - 26
Multithreading - 9, 373

**N**

not allowed characters - 86
numeric - 92, 100, 118, 375

**O**

ODBC - 26, 167, 189, 190
ODBO - 350
ODBO Connection - 352
order of creation - 87, 308, 309, 314, 339, 344, 346

**P**

Page selector - 75, 125, 126, 144
Palo Error - 27, 171
Palo ETL Server - 214
Palo is cell-related - 52
Palo ODBO provider - 350
Palo ODBO Provider - 353
Palo Service Manager - 22, 49, 366
Palo Settings - 22, 23
Palo toolbar - 156
Palo Wizard - 224, 226, 280, 281, 285
PALO.AFILTER - 307, 308, 333
PALO.CUBE_LIST_DIMENSIONS - 380
PALO.DATA - 54, 62, 73, 74, 79, 124, 130, 137, 139, 140, 198, 199, 204, 212, 255, 367, 376, 380
PALO.DATABASE_LIST_CUBES - 380
PALO.DATABASE_LIST_DIMENSIONS - 380
PALO.DATAC - 54, 62, 73, 74, 79, 124, 137, 139, 140, 142, 198, 199, 204, 211, 212, 255, 367, 376
PALO.DATAT - 130, 367, 376
PALO.DATAV - 62, 376
PALO.DATAX - 377
PALO.DDATABASE - 377
PALO.DFILTER - 307, 308
PALO.DFILTER() - 334, 335
PALO.DIMENSION_LIST_CUBES - 380
PALO.DIMENSION_LIST_DIMENSIONS - 381
PALO.EADD - 176, 182, 183, 185, 368, 377
PALO.EALIAS - 381
PALO.ECHILD - 377
PALO.ECHILD_COUNT - 377
PALO.ECOUNT - 378
PALO.EDELETE - 186, 188, 377
PALO.EFIRST - 377
PALO.EINDEX - 378
PALO.EINDEX - 378
PALO.EINDENT - 377
PALO.ELEVEL - 378
PALO.ENAME - 378
PALO.ENEXT - 378
PALO.EPARENT - 378
PALO.EPARENT_COUNT - 378
PALO.EPREV - 378
PALO.ERENAME - 378
PALO.ERROR_LOG - 215, 217, 378
PALO.ESIBLING - 379
PALO.ESIBLING - 379
PALO.ETOPLEVEL - 379
PALO.ETYPE - 379
PALO.EUPDATE - 379
PALO.EWEIGHT - 379
PALO.HFILTER - 308

This handbook has been personalized for NA - ohneanmeldung
palo.ini · 29, 30, 31, 32
PALO.PICKLIST · 307, 308, 309, 330
PALO.ROOT_LIST_DATABASES · 381
PALO.SETDATA · 139, 140, 141, 142, 161, 172, 185, 195, 213, 368, 378, 379
PALO.SETDATAIF · 379
PALO.SORT · 307, 308, 309, 344, 349
PALO.SUBSET · 307, 313
PALO.TFILTER · 307, 308, 328, 331
parameters · 32
password · 31, 191, 262, 263, 280, 281, 282
Paste Data Function · 26, 79, 80, 136, 169, 196, 198, 203, 209, 210, 211, 215, 242, 246
Paste Elements · 26, 63, 71, 72, 74, 97, 105, 106, 186
Paste Function · 169
Paste horizontally · 75, 78, 81
Paste vertically · 75, 77, 81
Paste View · 26, 27, 60, 69, 70, 71, 72, 75, 80, 101, 110, 112, 114, 119, 120, 122, 125, 126, 128, 133, 134, 144, 157, 175, 201, 242, 251
path · 28
Pick list · 64, 103
Pick list filter · 308
Pivot Tables · 350
poweruser · 270, 271, 279, 280
push rules · 132, 134, 137, 213, 375

R

rights · 24, 61, 278, 280, 284
rights modifying · 280
Row titles · 62, 70, 108, 114, 125, 126, 144, 145, 157, 192, 201, 251
rule · 278, 286, 287, 288, 289, 290, 291, 292, 293, 335
Rule Editor · 287
Rule Editor (Advanced) · 288
Rule Editor (Advanced) · 288, 293, 295
Rule-Editor · 85
Rule-Editor (Advanced) · 9
rules processing · 298

S

Search & Select · 76
search attributes · 105
search elements · 98, 99
Select all · 66
Select branch · 66
selection-buttons · 65
server · 15, 24, 30, 32, 40, 45, 47, 73, 79, 142, 183, 193, 224, 229, 259, 260, 261, 264, 265, 267, 280, 285, 375, 376, 377, 378, 379, 380, 381
Server · 379
Server Browser · 44
server registration · 224, 285
setup · 11, 15, 17, 22, 166
Show all selection tools · 67, 105
Show element selector on double-click · 69, 72, 119
single workstation PC · 11
snapshot · 26, 82, 83
Sorting of dimensions · 87
Sorting of elements · 87
Sorting using ASCII-Code · 319
splash · 140, 149, 152, 153, 154, 278, 279, 280, 375, 379, 382
splash parameter · 149, 152, 153, 154, 382
splashing · 148, 158, 278
string · 127, 368
structures · 93, 125, 126, 143, 166, 192, 226, 242, 283, 374, 377
Subset Editor · 9, 309, 326, 330, 332, 334, 340, 343, 347, 349
Subsets · 307
system cube · 276
system-cubes · 274

T
text · 91, 92, 94, 97, 100, 101, 106, 109, 118, 124, 172, 183, 200, 231, 260, 368, 375

U
undo · 9, 156
undo all · 156
undo one operation · 156
user · 16, 31, 61, 262, 263, 280, 285
User management cubes · 61, 275, 277
user name · 31, 280, 281
user rights · 24, 224, 280

V
viewer · 270, 271

W
Windows-Installation · 10
Work Offline · 84
Worksheet Server · 263
Worksheet Server Application · 261
Wrap labels · 70

Z
Zero suppression · 69

This handbook has been personalized for NA - ohneanmeldung